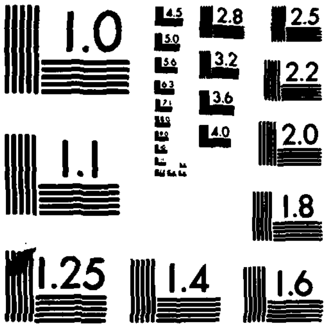
AD-A138 167 MODIFICATION TO THE ADVANCED DEGREE REQUIREMENTS 1/2
INFORMATION SYSTEM(U) AIR FORCE INST OF TECH
WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING
UNCLASSIFIED G P RANALLO DEC 83 AFIT/GCS/MA/83D-7 F/G 5/2 NL


1.0
4.5
5.0
5.6
6.3
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.1
1.8
1.25
1.4
1.6

MODIFICATION TO THE ADVANCED DEGREE REQUIREMENTS INFORMATION SYSTEM

THESIS

AFIT/GCS/MA/83D-7

Gene P. Ranallo
Major USAF

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

1

MODIFICATION TO THE ADVANCED DEGREE REQUIREMENTS INFORMATION SYSTEM

THESIS

AFIT/GCS/MA/83D-7

Gene P. Ranallo
Major USAF

AFIT/GCS/MA/83D-7

MODIFICATION TO THE ADVANCED DEGREE REQUIREMENTS INFORMATION SYSTEM

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

by
Gene P. Ranallo, B.S.
Major USAF
Graduate Computer Science

December 1983

Approved for public release; distribution unlimited

PREFACE

The Advanced Degree Requirements Information System (ADRIS) is an information and management tool converted for use at the Air Force Institute of Technology (AFIT) by Capt Matthew Waldron as a Master's Thesis in 1977. Since then there have been changes in policy at Hq USAF/MPPE pertaining to Academic Specialty Codes (ASC) and uses for these codes. New major commands and special operating agencies have also become operational since 1977. These changes caused erroneous information in ADRIS reports that needed to be corrected. ADRIS users wanted modifications to the system to make it easier to use, make the reports easier to understand, and to provide additional academic information for future educational planning. During this thesis, ADRIS was updated to correct information in the reports, enhanced to make it easier to use and understand, and expanded to include more officer records with specific Air Force Specialty Codes.

This thesis provided an opportunity to establish requirements for a system by working with users and incorporating their requirements into an operational system. The personal experience gained while doing this project will be invaluable in future job assignments as a personnel data systems manager at Hq AFMPC. This project also provided a chance to thoroughly analyze a system and then modify the system to correct existing errors. Fortunately, experience with the Air Force personnel system provided the background needed to perform requirements analysis for the ADRIS modification.

I thank Dr Henry Potoczny, thesis advisor, for his outstanding support. His contributions in analyzing FORTRAN code and offering countless suggestions when parts of the project seemed stymied were extremely valuable to this thesis. Mr Joe Hamlin, AFIT/ADO, provided superior technical advice on many items related to the CDC Cyber 74 computer. Capt James Moore, AFIT/EDP and Dr Charles Bridgman, AFIT/ENP established the system requirements used as a basis for this thesis, and their support and assistance was tremendous. Capt Rob Milne, AFIT/ENE, provided excellent assistance on artificial intelligence concepts and as a thesis reader.

Mr John Gates, Air Force Data Services Center, also provided important data changes that needed to be incorporated into ADRIS to update it.

My wife Rosie, daughter Lisa and son Kevin also gave me their full support and underwent a unique hardship during this time so that this project could be successfully completed.

Gene P. Ranallo

Table of Contents

Preface ii

List of Figures vi

List of Tables vii

Abstract viii

Chapter 1 IMPROVEMENTS TO THE ADVANCED DEGREE REQUIREMENTS INFORMATION SYSTEM 3

1.1 INTRODUCTION 3
1.2 OBJECTIVES 6
1.3 ORGANIZATION 10

Chapter 2 ADVANCED ACADEMIC DEGREE (AAD) MANAGEMENT SYSTEM 12

2.1 AIR FORCE SUPERVISOR 15
2.2 MAJOR COMMAND/SPECIAL OPERATING AGENCY 19
2.3 HQ USAF 20
2.4 ACADEMIC SPECIALTY CODES 21
2.5 ACADEMIC EDUCATION LEVEL 25

Chapter 3 SUMMARY OF ADRIS PROGRAM OPERATIONS BEFORE MODIFICATIONS 26

3.1 PROGRAMS SPLY AND DMND 26
3.2 PROGRAM ADRIS 34

3.2.1 ADRIS 35

Chapter 4 ADRIS MODIFICATIONS 42

4.1 PROGRAM LIBRARY 42
4.2 IMPROVE USER FRIENDLINESS 46
4.3 BACHELOR'S DEGREE RECORDS 50
4.4 MULTIPLE ACADEMIC SPECIALTY CODES (ASCs) 53
4.5 MAJOR COMMAND CODE EXPANSION 58
4.6 DATA FILE REVIEW 62
4.7 REPORT MODIFICATIONS 71
4.8 DATA BASE ANALYSIS 74
4.9 PROGRAM DOCUMENTATION 74

Chapter 5 MODIFICATION SUMMARY AND VALIDATION 76

5.1 IMPROVING ADRIS EFFICIENCY 78
5.2 VALIDATION 79

Chapter 6 CONCLUSION 81

Bibliography 83

Appendix A GLOSSARY OF ACRONYMS 84

Appendix B USER'S GUIDE 86
B.1 Purpose of ADRIS 86
B.2 USING ADRIS 87
B.3 TERMINAL OPERATION INSTRUCTIONS 103

Appendix C MAINTENANCE GUIDE 106

C.1 Building New Data Bases 106
C.2 File Format and Structure 115
C.3 File Directory 123
C.4 Miscellaneous Notes 123

Vita 126

List of Figures

Figure		Page
1	Process to Establish AAD Positions	14
2	AF Form 1779	17
3	Analysis of Sample Academic Specialty Code	23
4	Leighton Diagram of SPLY Program	28
5	Leighton Diagram of DMND Program	32
6	Leighton Diagram of ADRIS Program	36
7	Directory Example for Requirement Data Base	41
8	Flowchart of Multiple ASC Capability	57
9	Leighton Diagram of Updated ADRIS Program	59
10	Test for Major Command Code Change	63
11	Binary Search Tree Created by Subroutine BLDTREE	69
12	Test for Change in ASC Generalization	70
13	Magnetic Tape Transaction Request	108
14	Card Deck Jobs for AFMPC Tape Tests	110
15	Sample Output from Tape Test Run	111
16	Data Base Creation Examples	113
17	GENERAL Data File Structure	119
18	AREA Data File Structure	120
19	AGGREG Data File Structure	121
20	CBPCODE Data File Structure	122
21	MAJCODE Data File Structure	122

List of Tables

Table		Page
I	Hq USAF Functional Manager for Educ Requirements	15
II	First Character of Academic Specialty Code	22
III	ASCs Associated with Bachelor's Degree Records	86
IV	Aggregate Codes for ASCs	93
V	ASC Exceptions to Generalization	94
VI	Obsolete/Replacement ASCs	96
VII	Career Area Codes	97
VIII	CBPO Codes	99
IX	Major Command Codes	101
X	Tape and Read Formats	117
XI	ADRIS Program Library	124

ABSTRACT

The Advanced Degree Requirements Information System (ADRIS), an interactive data retrieval system that resides on the CDC Cyber 74 computer system, was updated and enhanced. The system provides AFIT staff and faculty members information pertaining to Advanced Academic Degree job positions in the Air Force and to officers who possess advanced degrees. Changes were made to ADRIS to make it easier to use and to provide much needed enhancements.

Extensive testing was performed throughout the systems modification, and two primary users of the system were involved in evaluating the changes. The operational system was used as a basis for comparison tests to insure tally information in reports was accurate.

The program library containing all source code, data files, and procedure files was restructured to make it easier to use during future system enhancements or maintenance. A system user's guide and maintenance guide were revised to reflect all changes made to the system and to provide additional information not previously documented.

Chapter 1

IMPROVEMENTS TO THE ADVANCED DEGREE REQUIREMENTS INFORMATION SYSTEM

1.1 INTRODUCTION

The Advanced Degree Requirements Information System (ADRIS) was developed in 1974 at Gunter AFS, Alabama by Capt John Carmack. The initial purpose of this system was to support information requirements of educational planners at Air University, Maxwell AFB, Alabama. It was specifically designed to give statistical data on the positions requiring officers with Advanced Academic Degrees (AAD) and on the officers who possess AADs. This type of information was important to educational planners because of their continuing efforts to establish and maintain educational programs in the Air Force that would insure qualified officers were available to fill duty positions requiring education at the advanced degree level.

AAD programs are based on validated Air Force position requirements, and AFR 36-19, "Advanced Academic Degree (AAD) Management System" outlines the policies and procedures used for identifying, reporting and validating line officer positions where an AAD is needed to insure that duties and responsibilities can be properly performed.

ADRIS was designed as an interactive computer program that prompts a user to enter key information pertaining to education level, Academic Specialty Code (ASC), grade, Consolidated Base Personnel Office (CBPO) code, major command code, and Air Force Specialty Code (AFSC).

After codes pertaining to this information are entered, the program does a data base search and generates a tally report based on the user parameters entered. The report information is furnished in two categories; output pertaining to the inventory of officers who matched the input parameters and to the Air Force job authorizations that required officers meeting the qualifications specified by the parameters.

After ADRIS was operational at Gunter AFS, the program was made available to users at the Air Force Institute of Technology (AFIT) School of Engineering via the AUTODIN communications network. It became an important tool for the AFIT staff and faculty, but the system became inactive when Capt Carmack was reassigned to Air Force Manpower and Personnel Center (AFMPC) in 1976. Because of the keen interest in ADRIS at AFIT, magnetic tape copies of the program and latest data bases were obtained from Gunter AFS. On 28 May 1976 the School of Engineering submitted a Data Acquisition Requirement (DAR) to establish

ADRIS at Wright-Patterson AFB, so that it could be used by AFIT.

The DAR was subsequently approved in September 1976 and conversion of the ADRIS code to a code structure compatible with the Control Data Corporation (CDC) CYBER 74 computer was performed by Capt Matthew Waldron as a thesis project. The system was fully operational at AFIT in March 1977 when Capt Waldron turned it over to the AFIT School of Engineering Office of Academic Support. Since that time, ADRIS has been used by AFIT faculty, staff, and students to obtain information concerning advanced degree job requirements and the officers possessing advanced degrees. However, the use of ADRIS is currently limited to staff and faculty members.

ADRIS is a viable retrieval system that furnishes important information on a daily basis to the AFIT staff and faculty. Although it is a relatively simple system to use with the prompting designed by Capt Carmack and Capt Waldron, the user must be familiar with codes pertaining to education level, grade, CBPOs, and major commands. These codes are used to enter the input parameters and the codes must also be interpreted when reading the output reports from the program. For someone who uses ADRIS often and is familiar with the code structures and code values, it was relatively easy to input the code parameters but the reports were difficult to interpret. For first time users or users relatively inexperienced with Air Force data codes, entering parameters and interpreting the output was difficult. Therefore the system needed a higher degree of user friendliness so it could better support old and new users.

There were also other system modifications suggested by Capt James Moore, AFIT/EDP and Dr Charles Bridgman, AFIT/ENP. They were the primary users of ADRIS, and Dr Bridgman was responsible for building new ADRIS databases each quarter using data tapes furnished by Hq AFMPC. In addition, during an extensive review and analysis of ADRIS, other problems were detected related to erroneous output caused by old code structures and data tables in the system and these had to be corrected to insure system output was accurate.

1.2 OBJECTIVES

The following objectives were established for this modification to ADRIS:

(1)Increase user friendliness by adding new on-line documentation and expanding existing documentation. In addition, review user requirements for a processor that can accept text input for base, major command, grade, and education level codes. Increase the visibility to the user of the career area option that is already available in the system. Search for other ways to enhance user friendliness by becoming thoroughly familiar with ADRIS and by seeking ideas from system users.

(2)Modify the education level parameter to include Bachelor's Degree and expand the inventory data base by adding officers with a Bachelor's Degree who possess specific ASCs. This change will greatly improve educational planning capability for the scientific AFSCs in the School of Engineering. However, ADRIS will continue to be primarily used for planning and decision making involving advanced degree job authorizations and officers who possess AADs. Therefore, the current system title ADRIS is still appropriate and will not be changed.

(3)Modify the ASC input structure so that multiple ASCs can be entered at one time, instead of only a single ASC for each query (the system can accept codes that represent a grouping of ASCs (aggregate) and this will not be eliminated or modified). This modification will reduce time spent entering queries when all parameter information except ASC is the same for the desired data base searches. Also modify ADRIS reports to summarize tallies for all ASCs when multiple specialty codes are entered.

(4)Change the major command code structure so that ADRIS uses a two character code instead of a single character code. The current structure that uses a single character code is obsolete because more major commands and operating agencies exist now than when ADRIS was first developed, and use of a single character code causes erroneous output for some queries.

(5)Thoroughly analyze data file GENERAL that contains ASCs that are specific to the last character (the fourth position) and update the table with new ASCs as necessary. A complete review of existing codes used in the data table must be accomplished to determine table accuracy. Since the fourth character is not used for most ASCs, it is replaced by a 'Y' unless the ASC is contained in table GENERAL.

This process increases the efficiency of ADRIS, but when new ASCs that require all four characters are added to the manpower system, ADRIS will automatically replace the last character of the ASC with a a 'Y' when building the requirements data base unless file GENERAL is updated to include the new ASC (more information on this process, call ASC generalization, is in chapter 3 under the heading PROGRAMS SPLY AND DMND. Also add on-line documentation for file GENERAL so the user can see which ASCs are excluded from fourth character generalization.

Also review and analyze data files AREA, AGGREG, and CONVRT (chapter 3 contains more detailed information on each data file). AREA contains career areas and the AFSCs each area represents. AGGREG contains the aggregate specialty codes and the individual ASCs that each code represents. CONVRT contains obsolete ASCs and their respective replacement specialty code. Each of these data files is unchanged for over six years, and it is important for the data in each one to be correct to insure accuracy in system reports.

(6)Modify ADRIS reports to make them easier to interpret. All output information is coded, but information such as CBPO, major command and grade can be translated to make reports easier to read. This change will make ADRIS reports easier to understand for users who are not familiar with Air Force codes for the data items specified.

(7)Review the data base structure and build procedures to determine if a new data base structure will contribute to increased efficiency in the existing program.

(8)Enhance documentation in the ADRIS programs to make them easier to understand, to aid program debugging, and make systems modifications simpler to accomplish in the future.

A significant part of the thesis effort involved developing a thorough understanding of the ADRIS computer program designed and developed by Capt Waldron. The main program and subroutines had general comments that stated what the main activity was in the program or subroutine, but there were not many comments that explained what specific parts of each program or subroutine did. This made ADRIS code hard to understand, and a detailed analysis had to be accomplished to determine how the program operated.

1.3 ORGANIZATION

In addition to the above objectives, sufficient information was provided on the AAD Management System and on ADRIS so the reader can use the thesis as a single information source about this system, including a revised user's guide and maintenance guide. However, there was no duplication of the excellent pages by Capt Waldron on the AAD management System or to providing a detailed explanation of how the old ADRIS works. Instead, a more general approach was taken to these two areas that should satisfy the information needs of everyone but a subsequent AFIT student who performs additional modifications on ADRIS. For a detailed explanation of ADRIS programs (prior to this modification), a thorough review of chapters 3, 4, and 5 of Capt Waldron's thesis should be done, followed by a thorough review of Chapters 3 and 4 of this thesis. The organization of this thesis is described below.

Chapter 2 summarizes the AAD Management System, as contained in AFR 36-19, 'Advanced Academic Degree (AAD) Management System', dated 23 March 1983. The AAD Management System provided the basic framework for designing and developing the ADRIS data base and retrieval system, and a basic understanding of AAD policies is essential for anyone who using ADRIS. The next chapter summarizes the ADRIS program (prior to modification), but as stated earlier a more detailed description of Capt Waldron's efforts is contained in his thesis which is available in the AFIT library and through the Defense Technical Information Center.

Chapter 4 contains the background, analysis, design/implementation, and verification for each modification made to ADRIS. Chapter 5 analyzes the impact of changes on the system and validates results. The last chapter provides results and conclusions concerning this thesis. The appendices are a user's guide and a maintenance guide for systems maintenance.

The responsibility of the maintainer is to rebuild the files quarterly using the data tapes provide by Hq AFMPC. The maintenance guide explains this and it continues to have information useful to anyone who is involved with ADRIS maintenance in the future. These guides were completely reorganized and updated to reflect new code structures and user techniques, where applicable.

Chapter 2

ADVANCED ACADEMIC DEGREE (AAD) MANAGEMENT SYSTEM

AAD programs in the Air Force are established based on validated requirements for officers with AADs at installations world-wide. The primary objective of this program is to insure academically qualified officers are available at all times to solve Air Force management and technological problems.

A key element in the AAD Management System is the ASC, which is a four character code that defines the academic field of study required for an authorized duty position (the ASC structure is fully explained in AFR 35-25, "Educational Classification and Coding Procedures" dated 13 December 1982). Another important item in the classification of AADs is the education level. The ASC and education level will both be explained at the end of this chapter.

First line supervisors in the Air Force are responsible for determining specific positions requiring officers with AADs. Sometimes this involves making a determination that specific positions within an organization require AADs. However, in most cases the supervisor should use the entire workcenter approach and consider graduate education resources in his/her entire workcenter, instead of looking at individual positions. Then the supervisor can decide the total number of advanced degree officers needed in the workcenter and align the AAD requirements with appropriate grades and specialty codes.

All AAD requirements should be based solely on the level and type of education needed to meet required professional and technical performance standards for the work center and position identified. Justification for AAD positions should not be based on (1)eaucational background of the officer occupying an existing position, (2)the current inventory of available officers who possess AADs in the ASCs required by a particular workcenter, and (3)the subjective judgment of the supervisor to achieve some theoretical enhanced capability.

In addition to involvement by supervisors throughout the Air Force, there are functional managers at major command/special operating agency levels and Hq USAF who have roles in operating the AAD Management System. Table I shows the functional manager structure at Hq USAF (Ref 2: 8). There is also the Air Force Education Requirements Board (AFERB) that sets the limit on AAD positions within each career area. Figure 1 is a flow diagram of the process used to establish an AAD requirement for an officer position in the Air Force (Ref 2: 7-15). The specific responsibilities of each agency involved in the AAD management process are described below.

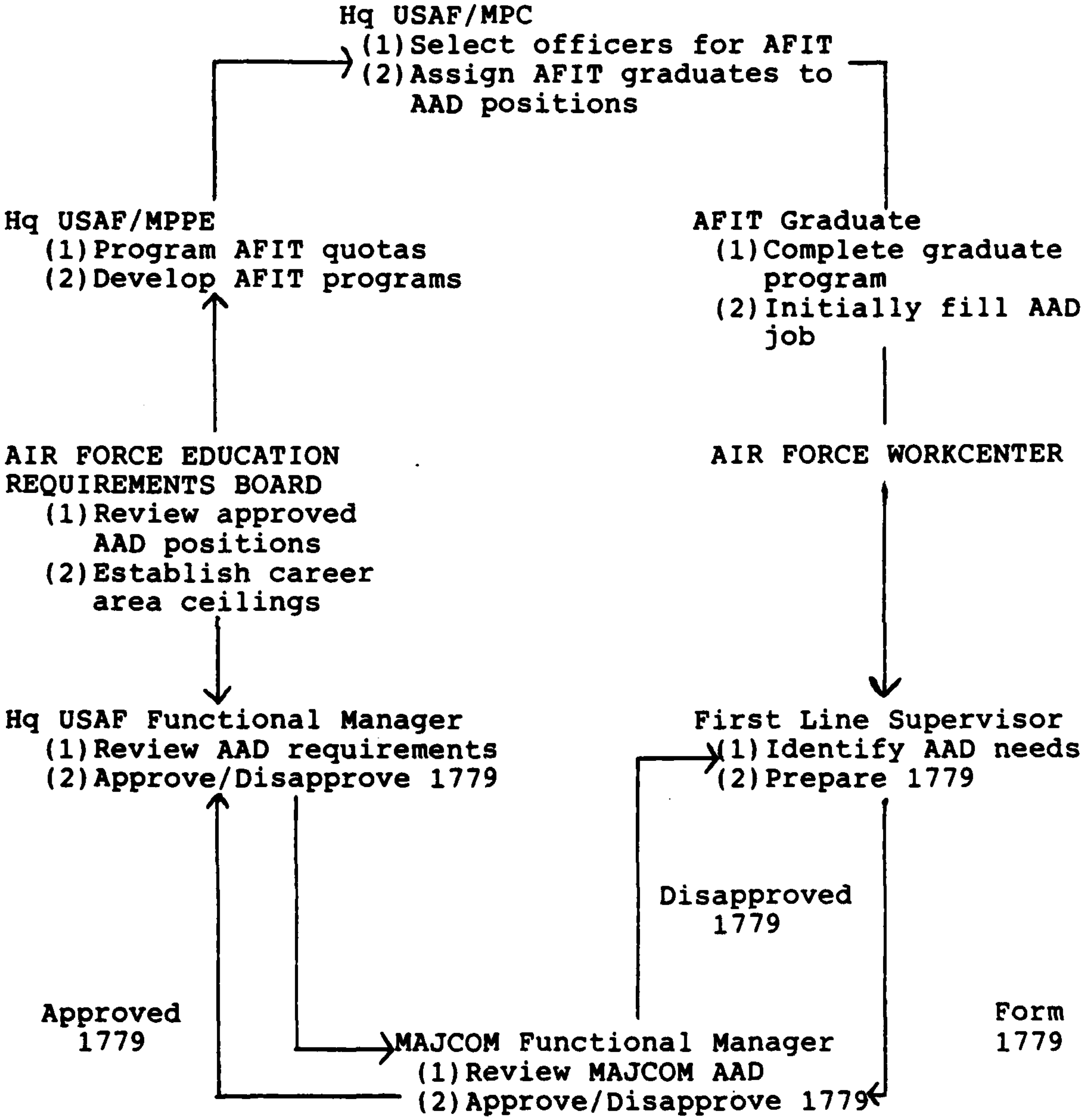

Hq USAF/MPC
(1)Select officers for AFIT
(2)Assign AFIT graduates to AAD positions
Hq USAF/MPPE
(1)Program AFIT quotas
(2)Develop AFIT programs
AFIT Graduate
(1)Complete graduate program
(2)Initially fill AAD job
AIR FORCE EDUCATION REQUIREMENTS BOARD
(1)Review approved AAD positions
(2)Establish career area ceilings
AIR FORCE WORKCENTER
Hq USAF Functional Manager
(1)Review AAD requirements
(2)Approve/Disapprove 1779
First Line Supervisor
(1)Identify AAD needs
(2)Prepare 1779
Disapproved 1779
Approved 1779
Form 1779
MAJCOM Functional Manager
(1)Review MAJCOM AAD
(2)Approve/Disapprove 1779

Figure 1, Process to Establish AAD Positions

Table I, Hq USAF Functional Mgrs for Education Requirements

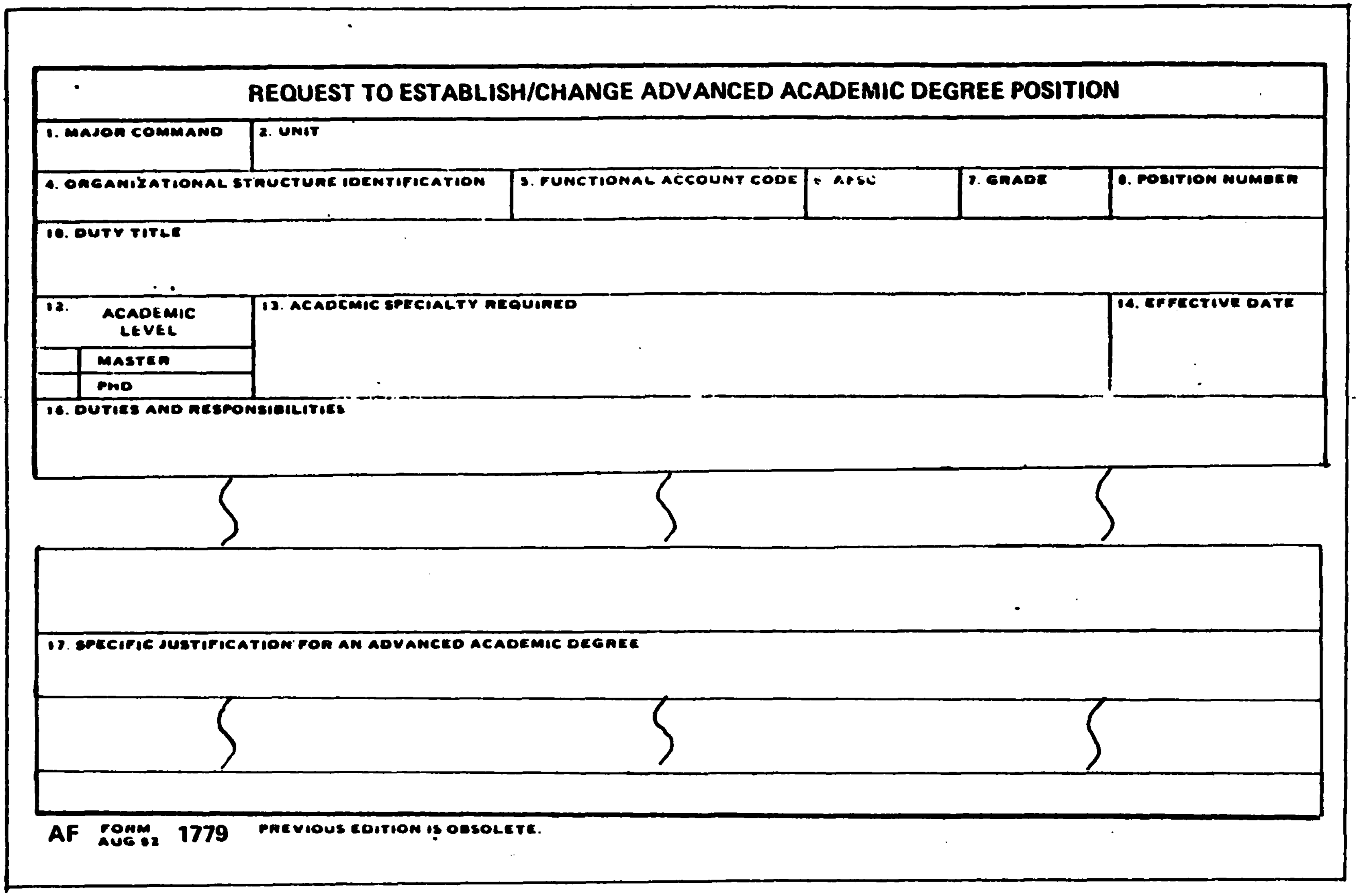
Career Area	Utilization Fields	Functional Mgr
Admin	70XX	Hq USAF/DA
Civ Eng/Svcs	55XX,62XX	Hq USAF/DE
Com-Elect	30XX	Hq USAF/XO
Comptroller	69XX,67XX,0056	Hq USAF/AC
Computer Tech	51XX,0960	Hq USAF/AC
Educ/Training	75XX,0900,0940,0950	Hq USAF/MP
History	0930	Hq USAF/CV
Intelligence	80XX,57XX,0910	Hq USAF/IN
Logistics	31XX,40XX,60XX,64XX,65XX 66XX,0046,0096	Hq USAF/LE
Manpower	74XX	Hq USAF/MP
Operations	10XX-19XX,21XX-23XX,0026,0036 0066,0076,0086,0216,0515	Hq USAF/XO
Pers Resources	0016,73XX	Hq USAF/MP
Public Affairs	79XX,87XX	SAF/PA
Sci/Dev Engr	26XX-29XX	Hq USAF/RD
Sec Police	81XX	AFOSP
Space Ops	20XX	Hq USAF/XO
Special Inv	82XX	AFOSI
Weather Avia	25XX,16XX	Hq USAF/XO

2.1 AIR FORCE SUPERVISOR

The supervisor is the individual who should be most familiar with duties and responsibilities assigned to positions under his or her control. Supervisors are required to survey each year all line officer unit manning document authorizations (grade of colonel and below) under their direct control. The object of this review is to determine in which positions it is important for the incumbent to possess an AAD to perform the prescribed duties.

The annual survey by the supervisor serves two purposes. First, the supervisor can review current AAD authorized positions and validate existing requirements for AADs. Secondly, the supervisor has the opportunity to identify new requirements within the workcenter where an AAD is considered essential for the duties to be properly performed. However, before requesting AAD positions the supervisor must consider available Air Force short course instruction as a means of fulfilling educational needs in the workcenter. Only when it is determined that short course instruction will not fulfill the educational needs of a workcenter should the supervisor submit the request for a new AAD position.

AF Form 1779, "Request to Establish/Change Advanced Academic Degree Position" (figure 2 depicts a sample of this form) is used to record new masters or PhD requirements for authorized Air Force line officer manning document positions, to request changes to AAD positions (for example, a different ASC for a validated AAD position), and to delete AAD requirements which the supervisor believes are no longer needed to perform duties in the position. Supervisors prepare AF Form 1779 and submit it through the command chain, with the Base Director of Personnel responsible for overseeing the AAD system at each Air Force installation. The AF Form 1779s must be submitted in time to process through base level channels so the major command functional managers will receive them no later than 1 September of each year.



REQUEST TO ESTABLISH/CHANGE ADVANCED ACADEMIC DEGREE POSITION

1. MAJOR COMMAND	2. UNIT			
4. ORGANIZATIONAL STRUCTURE IDENTIFICATION	5. FUNCTIONAL ACCOUNT CODE	6. AFSC	7. GRADE	8. POSITION NUMBER
10. DUTY TITLE				
12. ACADEMIC LEVEL □ MASTER □ PHD	13. ACADEMIC SPECIALTY REQUIRED			14. EFFECTIVE DATE
16. DUTIES AND RESPONSIBILITIES				
17. SPECIFIC JUSTIFICATION FOR AN ADVANCED ACADEMIC DEGREE				

AF FORM AUG 82 1779 PREVIOUS EDITION IS OBSOLETE.

Figure 2, AF Form 1779

18. ORIGINATING OFFICIAL		19. REVIEWING OFFICIAL	
TYPED NAME, GRADE, TITLE, AND OFFICE SYMBOL		TYPED NAME, GRADE, TITLE, AND OFFICE SYMBOL	
DATE	SIGNATURE	DATE	SIGNATURE
FOR MAJCOM/HQ USAF USE			
FROM:		TO:	
FROM:		TO:	
COMMENTS OR REASON FOR DISAPPROVAL			

20. MAJCOM FUNCTIONAL MANAGER					21. HQ USAF FUNCTIONAL MANAGER R		
TYPED NAME, GRADE, TITLE, AND OFFICE SYMBOL					TYPED NAME, GRADE, TITLE, AND OFFICE SYMBOL		
SIGNATURE					SIGNATURE		
RECOMMENDATION					HQ USAF FUNCTIONAL MA		
DATE	□	APPROVED	□	DISAPPROVED	DATE	□	APPROVED FOR ENTRY ON MANPWER DOCUMENT
22. MAJCOM AADMS MONITOR					23. MAJCOM MANPOWER AND ORGA		
DATE	SIGNATURE				DATE	SIGNATURE	

AF Form 1779 (Continued)

2.2 MAJOR COMMAND/SPECIAL OPERATING AGENCY

Functional managers at the major command receive all AF Form 1779s pertaining to their functional area of responsibility that are submitted by the bases in their command. The major command functional managers are responsible for closely monitoring AAD changes within the command. Each major command functional manager has authority to disapprove requests for changing or establishing graduate degree requirements which are not adequately justified or not warranted; when this occurs, the AF Form 1779 is returned by the functional manager to the submitting base and subsequently to the originating supervisor. The major command functional manager submits all approved requests to the major command AAD Management System monitor (someone assigned to the Directorate of Personnel), and all requests are forwarded to the proper Hq USAF functional manager.

When approved actions are returned by Hq USAF, the major command functional managers furnish a copy of the approval to the originating base and also provide a copy to Manpower and Organization at the major command (Manpower and Organization is solely responsible for updating the manpower data system with the approved AAD requirement). For requests disapproved by Hq USAF, the major command functional managers must make certain the originating base is advised of the disapproval action.

2.3 HQ USAF

Functional managers at Hq USAF are in the best position to identify education requirements for their area of responsibility for the entire Air Force. In addition to extensive technical expertise and experience, their position at the Hq USAF level gives them the day-to-day knowledge needed to fully understand short, intermediate, and long range duty position objectives for their specific area of responsibility. AAD positions approved by the Hq USAF functional managers become Air Force validated AAD requirements. Disapproved requests are returned to the appropriate major command functional manager for disposition as stated earlier in this chapter. In addition to the Hq USAF functional managers, responsibilities are set forth in AFR 36-19 for other Hq USAF offices:

(1) Air Force Education Requirements Board - This board establishes the ceiling for AAD requirements within each career area. It consists of at least 11 members, including the Director of Personnel Programs, DCS/Manpower and Personnel, as the chairperson. This board is responsible for developing the Air Force position related to current as well as future AAD requirements. It convenes at least every two years (Ref 2: 9).

(2) MPPE - This office uses validated AAD requirements as the basis for developing educational programs for AFIT. It is also responsible for programming AFIT quotas based on forecasted requirements established by the major commands (Ref 2: 7).

(3) MPC - Hq AFMPC is responsible for selecting personnel to fill AFIT program quotas and also making sure that graduates of fully-funded advanced degree programs are initially assigned to positions that have been validated as requiring an AAD (Ref 2: 7).

2.4 ACADEMIC SPECIALTY CODES

The ASC is a four character code defining the academic field of study required for an Air Force duty position. The ASC structure was established so a code would be specific enough to insure an academically qualified officer is assigned to a position, and yet be general enough to allow for necessary flexibility in the Air Force assignment system. All ASCs are listed in AFR 300-4, ADE AC-030.

The first character of the code specifies the general academic field. There are ten numerical code values and each one represents a grouping of closely related academic fields. A listing of the codes and related academic fields is in Table II (Ref 1: 10). The second character defines major academic field.

This is an alpha character and this field generally identifies the academic degree awarded by a college or university. It is normally considered to be a persons' academic major or field of study. The third character of the ASC defines the academic specialization level. This alpha character represents a subdivision of the major academic field, and is normally equivalent to an academic concentration of at least three courses in the academic field. The fourth character describes the subspecialization level. When used, this usually represents a group of courses associated with a given specialization. However, subspecialization is usually not identified nor reported. Figure 3 contains an example that shows a complete breakdown of ASC "1ATA" (Ref 2: 5).

Table II, First Character of Academic Specialty Code

1st Character	Academic Area
1	Administration,Management,Mil Science
2	Arts,Humanities,Education
3	Biological and Agricultural Sciences
4	Engineering
5	Law
6	Mathematics
7	Medical Sciences (academic specialty codes related to this area are not in ADRIS data bases)
8	Physical Sciences
9	Soc!al Sciences
0	Interarea

When specializations are related to two or more general areas of study or academic fields, they are referred to as

ACADEMIC SPECIALTY CODE - 1ATA

1 - First character is numeric. It shows the general area of study. There are 10 areas that can be used. "1" means administration, management, and military science.

A - Second character is alpha. It shows major academic field, which is the academic degree awarded by a university or college. It is considered the major field of study for an individual. The "A" means business adminstration or management.

T - Third character is alpha. It shows the specialization or subdivision of the major academic field. It is generally equivalent to an academic concentration of at least three courses in the academic field. The "T" indicates transportation management.

A - Fourth character is alpha. It shows the subspecialization. Usually this level is not identified nor reported, but there are selected ASCs where it is designated.

Figure 3, Analysis of Sample Academic Specialty Code interarea specialization. If the interarea specialization is not identifiable as falling under only a general area of study, the first character will be "0". However, when the interarea specialization is not identifiable as falling under both a general area of study and a major academic field, the first two characters of the ASC will be "0Y".

The ASC should always match the assigned actual duties required in an Air Force position. Most of the time, the ASC relates directly to the AFSC authorized for the position. However, there are some cases where the ASC does not directly match the AFSC.

The ASC structure also uses general codes in addition to "0". The code "X" means "other", and it is used when reported specializations or subspecializations are not considered reasonable approximations of those listed in AFR 300-4. Code "Y" means "not applicable" or "none". When the academic specialty cannot be identified down to subspecialization, the missing spaces are filled with the code "Y". Code "Z" means "unknown". It should not be considered a permanent alpha character that remains in an ASC; a code "Z" must be removed as soon as the appropriate data pertaining to the ASC is located (Ref 1: 9).

2.5 ACADEMIC EDUCATION LEVEL

The education level is a single character code representing the level that an individual has achieved in his advanced academic education. Five codes are used to represent advanced academic education:

P - Master's Degree, fewer than 30 post graduate hours.
Q - Master's Degree, 30 or more post graduate hours, no Doctoral Degree
R - Doctoral Degree
2 - currently enrolled in AFIT Master's Degree program (officers in this category are not in ADRIS)
3 - currently enrolled in AFIT doctoral program

Education level is one of the selection parameters used by ADRIS, and code values "2" and "3" apply to students in an AFIT resident institute program or a civilian institution degree program (officers identified by code "2" are not included in the ADRIS Inventory data base).

Although five codes are used for advance education level, only codes "P" or "R" are needed in the manpower system since these two codes define masters and doctoral needs, respectively. These two codes, plus the other three codes referenced above, can all be used to represent an individuals education level in the personnel system.

Chapter 3

SUMMARY OF ADRIS PROGRAM OPERATIONS BEFORE MODIFICATIONS

All ADRIS programs are coded in FORTRAN and the system operates on the CDC CYBER 74 computer. As stated before, the primary purpose of ADRIS is to provide AFIT staff and faculty members an on-line, quick response capability to retrieve statistical data (in the form of numerical tallies) pertaining to AAD authorized positions and Air Force officers with advanced degrees. To simplify the explanation of how the ADRIS programs work, this chapter divides the system into two separate areas. These areas are the data base build programs and the programs used to retrieve specific tally information based on input parameters from a user.

3.1 PROGRAMS SPLY AND DMND

Two separate data bases are used with ADRIS. One database (Inventory) represents the supply of Air Force officers who possess an AAD, and the second data base (Requirements) contains AAD validated position requirements. Program SPLY builds the

Inventory file and DMND builds the Requirements file. The records needed to build these databases are furnished quarterly by Hq AFMPC in the form of two magnetic tapes. Data for SPLY comes from an end of month officer personnel file at Hq AFMPC and the data for the DMND program comes from an end of month manpower file at Hq AFMPC. A main program and five subroutines are used to build the inventory data base.

Records are read one at a time from tape. The first character of the ASC, which specifies the general area of study described in table II is used as a key to temporarily store each incoming record. Once all the records are processed from the tape, these separate files that are based on the ASC first character are sequentially merged together. The beginning and ending points of each area are permanently marked and are located using a pointer file that simplifies searches performed by the ADRIS retrieval program.

A description of each subroutine in the SPLY program follows, and figure 4 is a Leighton Diagram of the SPLY program. (Note: Leighton Diagrams will be used to show program flow. Rectangles are used to portray program modules (subroutines), with the main program appearing at the left side of each diagram and each subsequent module appearing to the right, in turn. The horizontal dimension shows the overall program hierarchy, while the scope of control is shown by the vertical dimension. Subroutine calls proceed from top to bottom

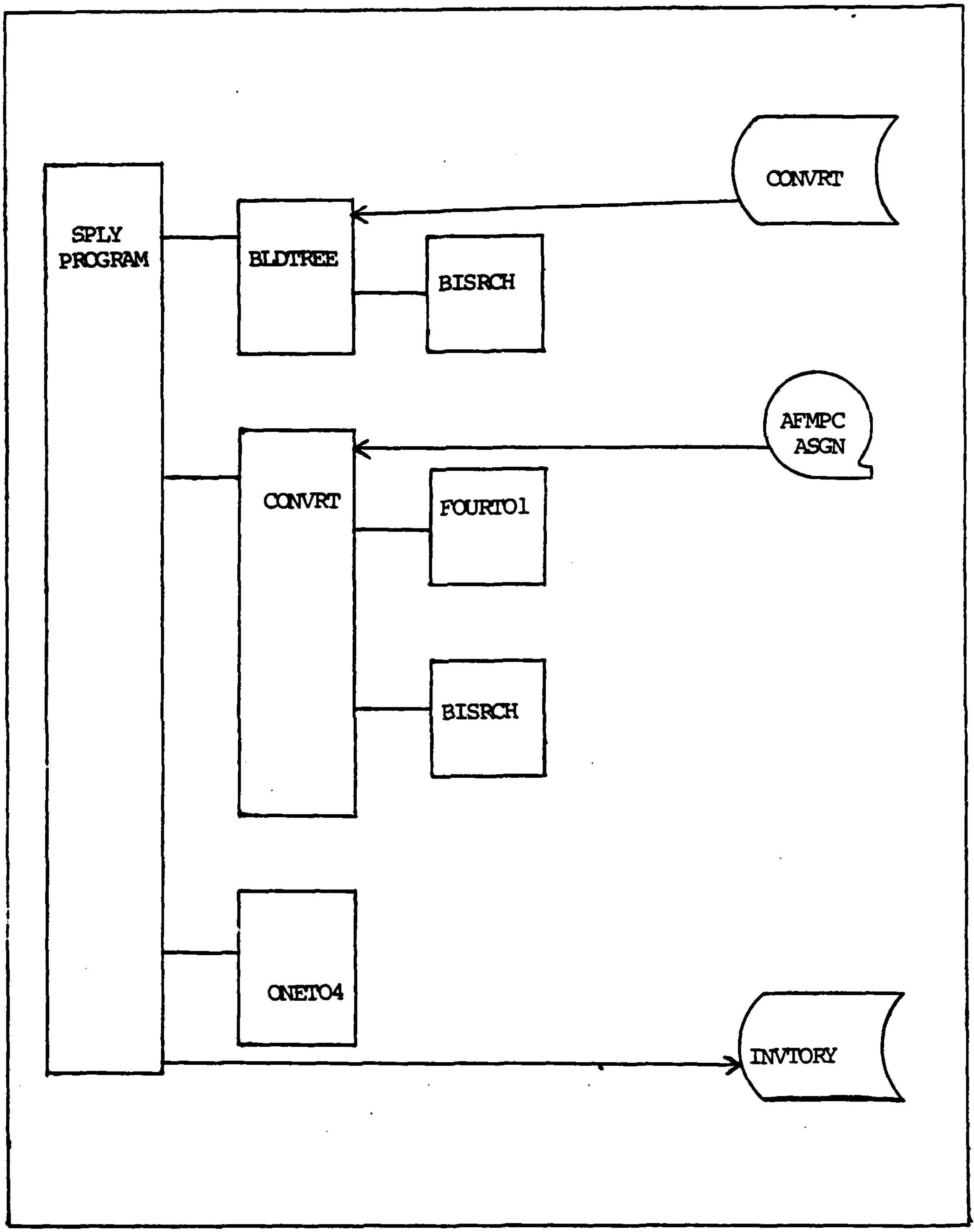


Figure 4, Leighton Diagram of SPLY Program and left to right (Ref 6: 45-46). In figure 4, program SPLY calls BLDTREE, and then BLDTREE calls BISRCH. When BLDTREE finishes, CONVRT is called next by SPLY. CONVRT receives tape supplied by Hq AFMPC, and then it calls FOURTO1 and BISRCH.

(1)BLDTREE - This subroutine creates and loads a binary tree used to identify obsolete ASCs and provide replacement ASCs. The obsolete ASCs and their respective replacement code are stored in an auxiliary data file titled CONVRT, and this file is used by BLDTREE to construct the tree used for checking each ASC contained on the tape file furnished by Hq AFMPC.

(2)FOURTO1 - This subroutine reformats ASCs read on the Hq AFMPC magnetic tape.

(3)BISRCH - This subroutine searches the binary tree to determine if ASCs on the tape file have matching codes in the tree. When a match is found, the ASC value is one that needs to be converted. If there is no match, processing continues to the next record on the tape file.

(4)CONVRT - When an ASC match is found in the BISRCH routine, CONVRT converts the obsolete ASC to the correct replacement value as specified in the binary tree. The replacement code becomes the value stored in the SPLY data base. This conversion procedure is necessary because there are over 2600 ASCs, and some codes eventually become obsolete and replacement codes are needed.

There is no regularly scheduled conversion of obsolete codes in the personnel or manpower files at Hq AFMPC, so as these obsolete codes become known by the AFIT staff or faculty, they can be converted to the replacement specialty. Updating the CONVRT data file with the obsolete and replacement code will accomplish this conversion during any database build performed after the data file is updated.

(5)ONETO4 - This subroutine is used to reformat ASCs.

A main program and eight subroutines build the Requirements data base. Some of the main concepts used to create the Inventory file also apply when building the Requirements file. Records are read one at a time from the magnetic tape furnished by Hq AFMPC, and the first character of the ASC for each officer record is used for sorting and subsequent record merging.

One concept extremely important in using program DMND is ASC generalization. During the mid-1970s, Hq USAF/MPPE determined that Air Force requirements (demand) were such that only a small number of specialty codes needed identification of the subspecialty in the fourth position. At that time, only 155 ASCs were identified in this category. In addition, MPPE decided that there were some requirements where specification of "X" in character position three of the specialty code could be generalized to reflect "Y". Initially there were 57 ASCs put in this category. The DMND program was designed to handle specialty code generalization, and this generalization procedure makes ADRIS operate more efficiently because most ASCs are generalized and this generalization procedure simplifies the retrieval process.

Subroutines BLDTREE, FOURTO1, BISRCH, CONVRT, and ONETO4 that are part of the SPLY program also support the DMND program and their tasks are identical to those performed in the SPLY program. A description of the other subroutines used in the DMND program follows, and figure 5 shows the DMND program.

(1)PUTGEN - This subroutine creates a hash table used to check the ASC in each record on the magnetic tape file from Hq AFMPC. The hash table is built using an auxiliary data file titled GENERAL and it contained 212 ASCs that were pre-identified as needing either the full four character ASC or the characters "XY" in the last two positions of the ASC.

(2)GETGEN - This subroutine gets the ASC from each incoming record on the magnetic tape and does a hash table look up for the ASC. If there is no match (ASC not found), this means the ASC can be generalized by replacing the actual fourth character with a "Y" so the ASC can be stored in the data base in generalized form. After a match either the ASC is stored in the database with no changes or the value "XY" is used to replace the third and fourth character of the ASC. Integer codes of "0" or "2" are used in the GENERAL data file to establish which action to take when a match does occur.

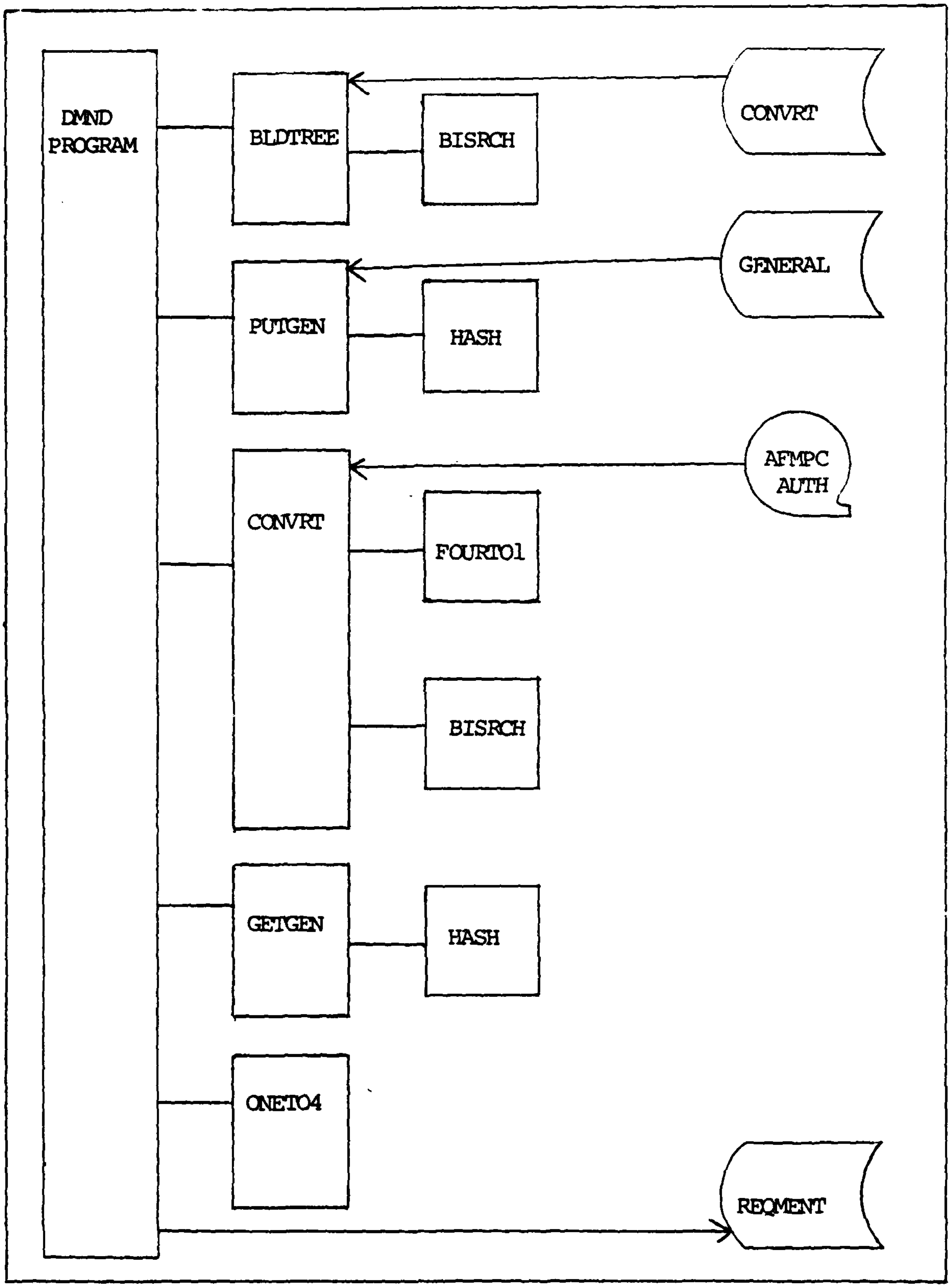


Figure 5, Leighton Diagram of DMND Program

(3)HASH - This subroutine is used to convert each ASC in the GENERAL data file to a hash value that is used in the hash table created by subroutine PUTGEN. Subroutine GETGEN also uses HASH to convert the ASC from each incoming record to a hash value that can be looked up in the hash table to determine if the ASC is contained in the table.

After the Inventory and Requirements files are built, they are used by the ADRIS retrieval program to respond to all queries. These files are static in that they remain intact, with no day to day changes, once they are created. They are updated only when new magnetic tapes are furnished by Hq AFMPC and the SPLY and DMND programs are executed again. These magnetic tapes are furnished by Hq AFMPC once each quarter.

Each file contains the following information for each record:

(1)Education level

(2)ASC

(3)AFSC (prefix and suffix included, when applicable)

(4)Grade

(5)CBPO Code

(6)Major Command Code

For the Inventory database, each record containing these items represents an officer resource. In the Requirements file, each record containing these six data items represents a validated position requiring an officer with an advanced academic degree.

3.2 PROGRAM ADRIS

This program accepts the input parameters provided by the user, searches the data bases for the requested information and formats the standard report and any optional report formats selected by the user. An overlay structure is used for the ADRIS program, which means it is divided into sections called overlays that reduce the amount of memory required for job execution. By using overlays, different parts of the program occupy the same storage locations at different times, with each part containing data and instructions used at different times during job execution (Ref 3: 6-1).

Capt Waldron designed and coded program overlays to reduce memory requirements and resource usage, to increase query response times,and to allow future program growth. The overlays were needed because of the 60k limitation on CDC CYBER 74 interactive programs, and ADRIS nearly exceeded this limit before the overlays were used. However, memory requirements were reduced to 43K by using the overlays.

ADRIS source code was also available in non-overlay form, which is how the system was initially designed by Capt Waldron. Some subroutine names for the non-overlay form differ from those used in the overlay structure. The following program description is for the overlay version of program ADRIS, and the subroutine names are compatible with that version. Figure 6 depicts the program flow.

3.2.1 ADRIS

This is the root overlay, and it is the main driver program. It sets up the initial prompting to the system user, opens the Inventory and Requirements data bases, and opens the pointer file used to locate records based on the first position integer value of ASCs. The driver module passes control to two other overlays used in ADRIS. One is named BASIC and the other is SUMRY.

BASIC

This module controls gathering and storing of input parameters, performs data base searches for the parameters specified, and prints the output tally pertaining to the officer inventory and the job requirements. Three subroutines in BASIC support these tasks.

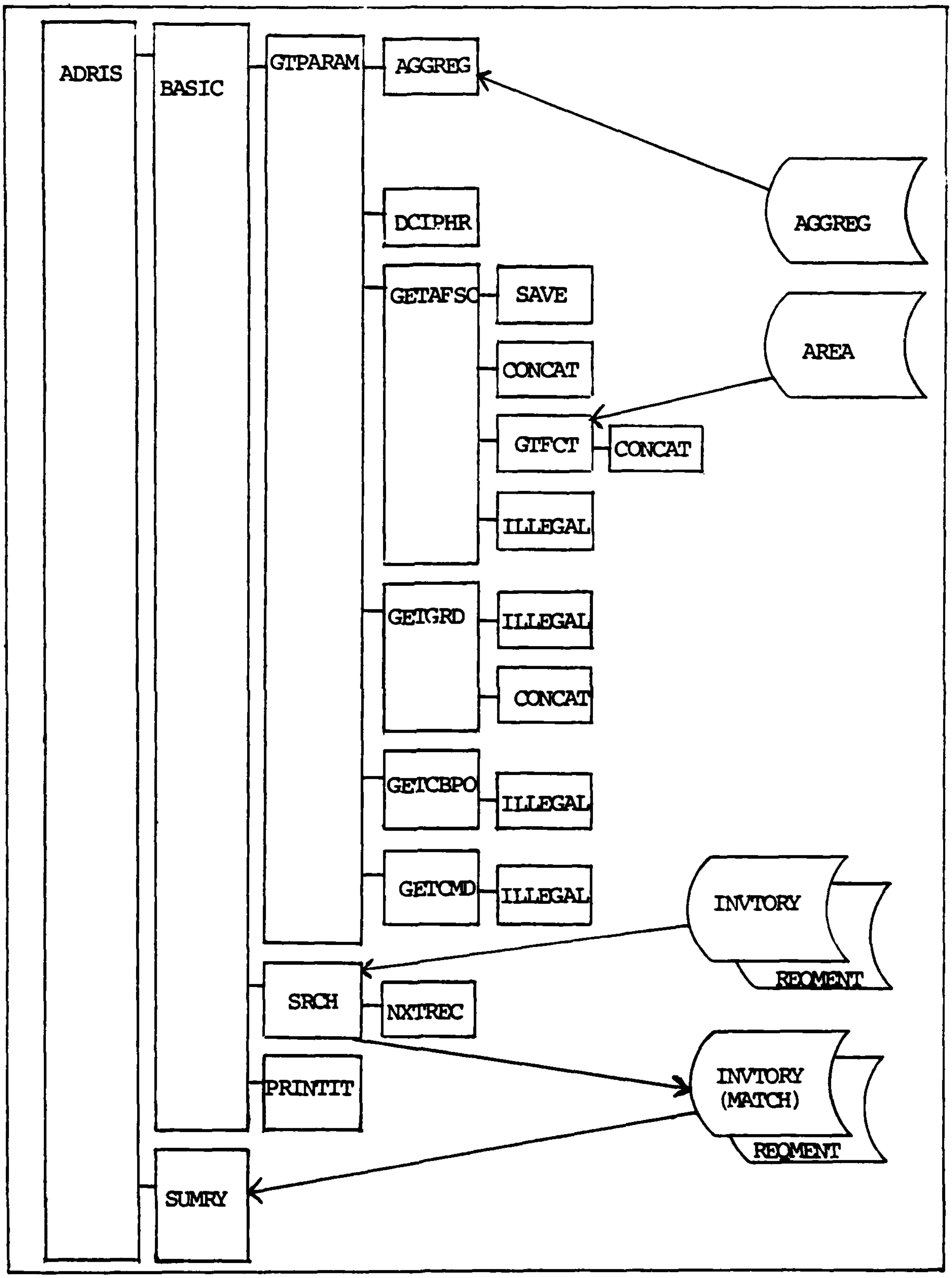


Figure 6, Leighton Diagram of ADRIS Program

(1)GTPARAM - This subroutine gathers and stores all data parameters entered by the user. It captures the desired parameter for education level (Master's Degree, Doctoral Degree, or currently enrolled in AFIT doctoral program) and the specialty code when a single ASC is entered. Program logic used for gathering and storing the parameters identified below is in GTPARAM and the supporting subroutines specified:

(a)Aggregate Specialty Code - When an aggregate code is detected by GTPARAM, subroutine AGGREG interprets the aggregate code and provides the individual ASCs that the aggregate code represents. The individual ASCs and associated aggregate codes are stored in data file AGGREG, which is used by subroutine AGGREG. This subroutine stores specific ASCs until they are needed for the data base searches.

(b)AFSC - Subroutine GTPARAM accepts the initial input for individual or multiple AFSCs, or for a career area that defines multiple AFSCs. After the AFSCs or career area are entered, subroutine GETAFSC is called and it stores the AFSCs. It can also store AFSCs specified by range (73XX to 79XX) or by general category (73XX means values 7300 to 7399). GETAFSC calls subroutine SAVE, which is used to set up storage for the specified AFSCs. GTPARAM also calls subroutine GTFCT when a career area is entered. GTFCT converts the career area to specific AFSCs, and then these are stored for use during the data base searches; the conversion is done using data file AREA, which identifies each career area and the AFSCs each area represents. GETAFSC also calls subroutine ILLEGAL, which provides an error pointer on the CRT screen to erroneous AFSC parameter input. Subroutine CONCAT is also used by GETAFSC. Its purpose is to concatenate characters from multiple separate computer words into a single word.

(c)Grade - GTPARAM can accept individual or multiple grades; valid grades are second lieutenant through colonel, codes O1 to O6. Subroutine GTGRD edits and stores the grades, and it stops storing grade data if a duplicate grade value is entered. It calls subroutines CONCAT and ILLEGAL. CONCAT performs the same task as it does in GETAFSC, and ILLEGAL creates an error pointer that informs the user of erroneous grade parameters.

(d)Base Identifier - This is the CBPO code. A single code or multiple codes, each two characters in length for up to a full line, can be entered. GTPARAM accepts the input and subroutine GETCBPO stores the CBPO ID codes. ILLEGAL is also used by GETCBPO to point at input data on the CRT screen that is in the wrong format.

(e)Major Command Identifier - This is the MAJCOM code. Single or multiple command codes (up to one full line) can be entered. GTPARAM accepts the input and subroutine GETCMD stores the command code. A single character code is required by the user, and GETCMD adds a leading zero to form a two position code, which is stored.

Subroutine GTPARAM will also accept "ALL" or "*" as input for these parameters. These values have special meanings to the program and when they are used the calls to the subroutines that support GTPARAM are not made for the specific code involved.

One other subroutine, DCIPHR, is called by GTPARAM. It is used to store single ASCs that are in the correct format and also sets a flag value that establishes the specific level of the users specialty code request (first, second, third, or fourth position). This flag is subsequently checked during the data base search.

(2)SRCH - This module extracts the Inventory and Requirements records from their respective data bases and compares the data in each record with the users selection criteria. When a record matches the users selection criteria, the tallies for the standard output report are incremented and the matched record is also stored for later use when creating special reports requested by the user. The first character of each ASC is the key element in the data base search, since the data bases are structured so that records within each file are merged together based on the ASC first character value.

A storage location directory is maintained for each data base, and it contains the starting location of each first character ASC value in the data base. By merging all first character values that are equal into the same area of the file, the search area in each data base (for a specific ASC) is greatly narrowed. Figure 7 shows a directory example from the requirements data base.

SRCH calls subroutine NXTREC, which reads groups of 100 records at a time into memory from the data base and unpacks the records one at a time, so the appropriate data in each record can be matched with the users' selection criteria to determine if the record should be selected.

(3)PRINTIT - This subroutine prints the header information for the standard output report to the user, prints tallies from the Inventory and Requirements parts of the query, and calculates/prints the Inventory to Requirements ratio. PRINTIT provides a tally for colonels (grade O6), but this information is not included in the calculated ratios.

SUMRY

In addition to the standard report format that ADRIS furnishes for all queries, six special report formats are available. Subroutine SUMRY calculates and prints the results for each special report selected by the user.

```
PD( 1)=        1    Inter-Area
PD( 2)=     1317    Admin, Mgmt, Mil Science
PD( 3)=     4367    Arts, Human, Education
PD( 4)=     4734    Biolog & Agricul Science
PD( 5)=     4750    Engineering
PD( 6)=     7089    Civil Law
PD( 7)=     7095    Math
PD( 8)=     7245    Phys Science
PD( 9)=     8159    Soc Science
PD(10)=     8746    YYYY ASCs
PD(11)=     9430    Aggregate ASCs
PD(12)=     9430    Last Rec + 1
```

Note: Directory references in left hand column range from one to twelve. These numbers do not equate to first char values given in table 2. In this example, Inter-area value "0" equals "1", Admin/Mgmt/Mil Sci value "1" equals "2", etc. Also, medical specialty codes (first char seven) do not appear in directory since medical specialties are not used in ADRIS. Special merge categories for YYYY specialty codes and aggregate codes are established in the directory.

Figure 7, Directory Example for Requirements File

Chapter 4

ADRIS MODIFICATIONS

This chapter documents each modification and enhancement made to ADRIS. Objectives outlined in Chapter 1 established the basis for most of these changes. Each modification or enhancement is divided into four areas: background, analysis, design/implementation, and verification.

A major change to ADRIS not identified in the original objectives involved the program library. This change has no impact on ADRIS operations, but it affects how future system changes are accomplished. This change involved modifications to existing program storage techniques, and an understanding of this new library structure is important for anyone who maintains or modifies ADRIS in the future.

4.1 PROGRAM LIBRARY

BACKGROUND - The original library for source programs, procedure files and data files was stored as an UPDATE file. There were eight programs, four data files, and five procedure files stored in the UPDATE library. UPDATE is a Network Operating System/Batch Environment (NOS/BE) utility on the CDC CYBER 74 computer used for editing programs stored in source program libraries on tape or disk. UPDATE is generally used to create and correct programs run as batch jobs that are too large to be handled by EDITOR (Ref 4: 26). When UPDATE is used, source files contain special control characters used by the utility. UPDATE was a useful library structure for Capt Waldron because his thesis involved a massive code conversion so ADRIS could operate on the CDC CYBER 74. He ran batch jobs using UPDATE and also used the text editor with interactive programming.

ANALYSIS - UPDATE was useful to Capt Waldron for his thesis work, but this utility seemed inefficient for the new modifications to be performed on ADRIS. Since the UPDATE utility and the original source library were not going to be used, another method for text editing and program storage was needed. Since the CDC CYBER 74 EDITOR was adequate for program modifications, it was selected as the text editor for this project. However, the programs and files in UPDATE form were not compatible with the EDITOR and therefore the program library was restructured to standard 80 character format, with all control characters from UPDATE deleted from the files. Another important consideration during the ADRIS modification was availability of the operational system.

Since the system would be continually used during the modification, no changes would be made to the operational system and no development work would be performed under the operational account number. This concept avoided two potential problems. First, ADRIS users would not need to be concerned with erroneous queries during development of the new system. Secondly, an accurate operational system would always be available to compare query results with the test system.

DESIGN/IMPLEMENTATION - The objective was to break the UPDATE library into separate programs so each one could be identified and compiled. Each was catalogued on the new CDC CYBER 74 account established to support developing a modified version of ADRIS. Procedure files and data files, which were stored separately in the ADRIS library and duplicated in the UPDATE library, were included in the new account.

Ten source programs, procedure files and data files were initially in the new library. After storing object code, generating data bases, creating new program TREE, and building new data files, the library consisted of 19 files. While this number of files is larger than the original ADRIS program library (18 files), the new library's structure eliminated duplication of all data files and of three overlay programs that was present in the old program library. In the new library, there is no file redundancy. The new library structure also gives a clearer presentation for ADRIS files and will make program maintenance and modification easier to perform in the future.

VERIFICATION - All source programs were compiled at least once before any modifications were made. This was done to insure each one was a valid source program and that all changes made to ADRIS by Capt Waldron during his systems development were properly catalogued in the source programs. After all programs were successfully compiled, the ADRIS data bases were created using these source programs and data files. The objective was to initially create a duplicate of the operational system on the new account and then test it against the operational system to confirm that the source programs were valid. This was accomplished by comparing the data base build output with the same output from the operational system (using the same files from Hq AFMPC) and then running the same queries on both systems and comparing the tallies.

During the verification process, one error in a file OPEN statement was found in the source program used for building the Requirements data base. After this was corrected, comparison queries between systems gave the same results. The new program library was the foundation for all modifications to ADRIS programs and files, and the new library structure became a permanent part of the system.

4.2 IMPROVE USER FRIENDLINESS

BACKGROUND - One objective was to analyze the requirement for user entry of translated data versus coded data, and to develop a processor for translated input if a requirement existed. After using ADRIS for many test queries and then discussing this objective with Capt Moore and Dr Bridgman, it was clear a text input capability to replace data codes was not essential for this system. Since there are only six different input parameters and many queries use the select (*) option for one or more parameters, text input would see little or no use by experienced ADRIS users. Access restrictions on this system coupled with infrequent new users also supported this position concerning text input. Therefore, input of trans- lated data was not a firm requirement and was replaced by other enhancements to aid all system users.

Capt Moore also wanted more visibility for the career area and the aggregate options. He had not used them before, but when he learned they were available in ADRIS he indicated the user should know this through on-line documentation.

ANALYSIS - User on-line documentation explaining options and input formats for each parameter was limited. This limited documentation may have been intentional to avoid delaying the user when query parameters were entered. However, the system was designed so that on-line documentation could be by-passed by a user who was familiar with ADRIS.

A decision had to be made after signing on to the system; if the user selected the documentation option, there previously was no way to discontinue it. This created problems for new system users as well old users who wanted to refresh their knowledge of ADRIS. After requesting on-line documentation, there was no provision for the user to later turn off the documentation and receive only minimal prompting for subsequent queries.

Another area where documentation was non-existent involved the use of CBPO codes and major command codes. These codes were in the user's manual, but an on-line capability seemed beneficial to a user needing quick access to the information to complete a query.

There was also a high level of user distraction over the display of start, stop, and search times for each query. This feature was useful for analyzing different query techniques, but as part of the output on every query it served no purpose and cluttered the display screen with useless information.

The career area and aggregate options were tested prior to establishing on-line documentation. Testing showed the options did not work (when any valid area code was entered, the system would respond with "ILLEGAL CAREER AREA DESCRIPTOR"; when aggregate codes were entered, the query results were sometimes erroneous). Capt Moore and Dr Bridgman confirmed this was possible since neither of them used these options before. Extensive analysis of the GETAFSC, GETFUNCT and AGGREG subroutines in overlay BASIC isolated the problems and modifications were completed that corrected them.

DESIGN/IMPLEMENTATION - On-line documentation was expanded by entering more comments for each parameter. The new comments provided more information about each parameter and gave clearer examples of the proper format to use for each parameter. To turn off on-line documentation (if used) after the first query was completed, an additional prompt was added that permitted the user to specify continue documentation or discontinue documentation on subsequent queries.

Documentation pertaining to CBPO and major command codes was added by creating data files that contained translations and codes for these two parameters. These files were designed to also support the requirement for translated data on all output reports (discussed later in this chapter). Extensive research involving Advanced Personnel Data System table 72 (MAJCOM Identity) ,table 305 (CBPO Identity), AFM 10-4, "AF Directory of Unclassified Addresses", and AFM 300-4, ADE CO-485 were needed to gather information for these new files. Information was also obtained through discussion with Mr Charles Zabel, AFMPC/MPCDDS3. To aid the user in locating the specific codes, the translations were stored in alphabetical order in both tables.

Start, stop and search times were eliminated from the SRCH subroutine in overlay BASIC, and a new print line was added that displayed the ASC the system was searching for. This provided the user with a meaningful heading for each query, and the heading is especially helpful when multiple ASCs are entered by the user. The heading is displayed before each new query.

The career area option was corrected by tracing the user entered area code from the point of input until it was processed by subroutine GETFUNCT. Tests showed the area code was lost prior to being passed to GETFUNCT. Additional tests isolated the area of subroutine GETAFSC where the code was lost. The actual problem was traced to an erroneous ENCODE statement where the same variable was used as input and output (Ref 5: 105-115).

The error was corrected by adding a new variable to subroutine GETAFSC and using it as input to the ENCODE statement. After this correction was made, documentation was added that briefly explained the option and offered the user a display of the area codes and their respective AFSCs from the data file AREA. The error in the aggregate option was found through extensive testing. The problem was traced to an erroneous format statement used to read data from file AGGREG. Source parameters in file AGGREG were also erroneous and these were corrected.

VERIFICATION - These changes were validated by performing numerous queries and using all possible combinations of requesting documentation, suppressing documentation, and specifying no documentation after it was initially used. Test results confirmed the documentation was displayed at the correct places and that turn on and turn off points were accurate.

CBPO and major command translations/codes were accurately displayed, as were career area codes/AFSCs and aggregate codes/ASCs. Print headings identifying the ASC for each query appeared at the correct point and the ASC was correct from one query to the next when multiple ASCs were used.

4.3 BACHELOR'S DEGREE RECORDS

BACKGROUND - ADRIS was designed to support educational planning related to advanced degree authorizations and officers with advanced degrees. There were no records on the officer inventory tape furnished by Hq AFMPC where the academic education level was a Bachelor's Degree or Bachelor's Degree plus.

Dr Bridgman was vitally interested in Bachelor's Degree for planning purposes related to both undergraduate and graduate programs at AFIT. His requirement related to the Inventory data base only, and did not include adding new records to the Requirements file.

ANALYSIS - Adding all Bachelor's Degree records to the existing data base would nearly triple the size of the file. This would add significant overhead to ADRIS and increase the overall search times because of the larger Inventory data base. Further analysis of this requirement revealed that Capt Moore was not currently interested in Bachelor's Degree information. He did point out that Bachelor's Degree records should not be included in the select (*) option data base search for education level, since this would distort the reports.

Additional discussion with Dr Bridgman resulted in a narrowing of the AFSC range for Bachelor's Degree records to 26XX-29XX, since these were the specific AFSCs of interest for him. This meant the data base growth would be very small, and there would be virtually no impact on search times. However, preliminary testing revealed that this range of AFSCs would not provide enough information to support planning needs. Therefore, the modification was redesigned so that Bachelor's Degree records were selected based on first character of ASC, with no regard to AFSC. ASCs beginning "0", "4", "6", and "8" were chosen, since these represent all specialties of interest to the School of Engineering.

The selection criteria on the inventory tape produced by Hq AFMPC had to be modified to add Bachelor's Degree records. Mr

Roy Adamson, AFMPC/MPCDMR, was contacted concerning this change and he agree to furnish a test tape with Bachelor's Degree records. A letter was submitted to his office to support the requirement.

DESIGN/IMPLEMENTATION - The test tape contained all officers whose highest academic education level was a Bachelor's Degree or Bachelor's Degree plus. To process this tape, the SPLY program required a modification so only Bachelor's Degree records with specific ASCs were added, along with the advanced degree records. This design would permit adding or deleting Bachelor's Degree records in the future by changing only a single line of code in program SPLY.

Program ADRIS also needed changes to access Bachelor's Degree records. Changes included the addition of education level "N" in subroutine GTPARAM ("N" represents bachelors degree). Subroutines SRCH and PRINTIT also needed changes. IN SRCH, code was added so records pertaining to Bachelor's Degree (code "N") and Bachelor's Degree plus (code "O") were selected when the user specified education level "N". SRCH was also modified to bypass a search of the Requirements data base when a query for Bachelor's Degree records was entered.

Subroutine PRINTIT was changed by modifying print headings so only information related to Inventory records would be displayed after a search for Bachelor's Degree records was completed. No changes were made to overlay SUMRY that prints the optional reports, since these reports were produced based solely on records selected during the data base search.

An important item for this modification involved the Requirements data base. No changes were needed in the DMND program used to build the Requirements data base. However, the ADRIS pointer file is affected anytime a new inventory tape is processed and the Requirements data base had to be rebuilt after the Inventory data base was created from the test tape furnished by Hq AFMPC. An unchanged requirements tape from Hq AFMPC was used to do this update (when new data bases are created, the Inventory file is always created first, followed by the Requirements file). More information on this process is in the maintenance guide.

VERIFICATION - A series of queries were run to show that Bachelor's Degree records could be retrieved. The results were checked to insure only ASCs "0", "4", "6", and "8" were contained in output reports for Bachelor's Degree queries. Test queries using the select (*) option for education level were also run to confirm that Bachelor's Degree records would not be selected.

4.4 MULTIPLE ACADEMIC SPECIALTY CODES (ASCs)

BACKGROUND - In the original design of ADRIS, the user could enter a single ASC or a single aggregate code representing pre-defined ASCs (tailored for use on the Requirements data base). The aggregate option has not been used at AFIT, so most queries involved either entry of a single ASC at a time or specifying the select (*) option which resulted in no restrictions on selection of ASCs.

The select (*) option generally gives excess data, and response time increases because of the search magnitude (when a specific ASC is used, the data base search is restricted to an area of the data bases containing records matching the first character in the ASC). Individual queries with single ASCs provided the type output most often desired, but when a user had more than one ASC to enter all the other parameters had to be reentered.

Computer resources needed to input these parameters over and over when they did not change was not significant, but the overall time needed by the user to get the desired output was increased because of the time spent reentering the same parameters. Capt Moore requested ADRIS be modified so that multiple ASCs could be entered whenever the user had a series of queries to run that involved different ASCs with the same parameter values for education level, AFSC, grade, CBPO, and MAJCOM. He also wanted the standard and optional reports to summarize tallies for all multiple ASCs entered.

ANALYSIS - Input and processing of ASCs was handled by program ADRIS (the root program) and subroutines GTPARAM and DCIPHR in program overlay BASIC. Two approaches were considered for this modification. The first involved redesigning the operating sequence of ADRIS whereby all ASCs entered by a user would pass to the SRCH module and then a loop would handle the ASCs individually. This concept would have required a complete redesign of the linking structure in program ADRIS that handled the PRINTIT subroutine (produces the standard output report) and the SUMRY overlay (produces the six optional reports).

In addition subroutine DCIPHR was being used to add a decode value to each ASC and set other parameters that made the search process more efficient. This decode value and the parameters were critical to the efficient operation of the data base searches, and this subroutine would have to be completely redesigned to provide the same information to multiple ASCs.

This design approach involved modifying two overlays and three subroutines and it would require nearly a complete redesign of the flow of data and sequence of operations in ADRIS. Therefore a less complex design was sought to implement this modification.

A design was needed that permitted the system to operate with the same pattern of data flow and sequence of operations while providing the user the capability to enter multiple ASCs when desired. After extensive analysis, a design plan was developed that achieved the objective of multiple ASC capability while maintaining the system basically in its current form.

DESIGN/IMPLEMENTATION - The design involved changes to root program ADRIS as well as modifications to subroutines GTPARAM and DCIPHR in overlay BASIC. A change was also needed in subroutine SRCH to allow tally information to accumulate when multiple ASCs were entered. No changes were needed in subroutines PRINTIT or in overlay SUMRY, whereas both would have needed extensive changes under the initial design concept considered. Under the original version of ADRIS, single ASCs were captured in subroutine DCIPHR. After the modification, subroutine DCIPHR is only used to provide decode values and to set parameters for each ASC prior to it being passed to subroutine SRCH.

The design required a new subroutine, GETASC, which captures the select (*) option, single ASCs, or multiple ASCs entered and then stores this data entered by the user for subsequent use in the SRCH subroutine. GETASC keys on spaces, characters, and commas while performing a series of validation and error checks to detect erroneously structured ASC codes.

Since ADRIS uses overlays, the COMMON area was expanded to save parameter values for subsequent reprocessing with different ASCs. Approximately 50% less lines of code were used to implement the modification using the design chosen versus the first design considered. A general flowchart is in figure 8 showing basic logic and design structure. A new Leighton

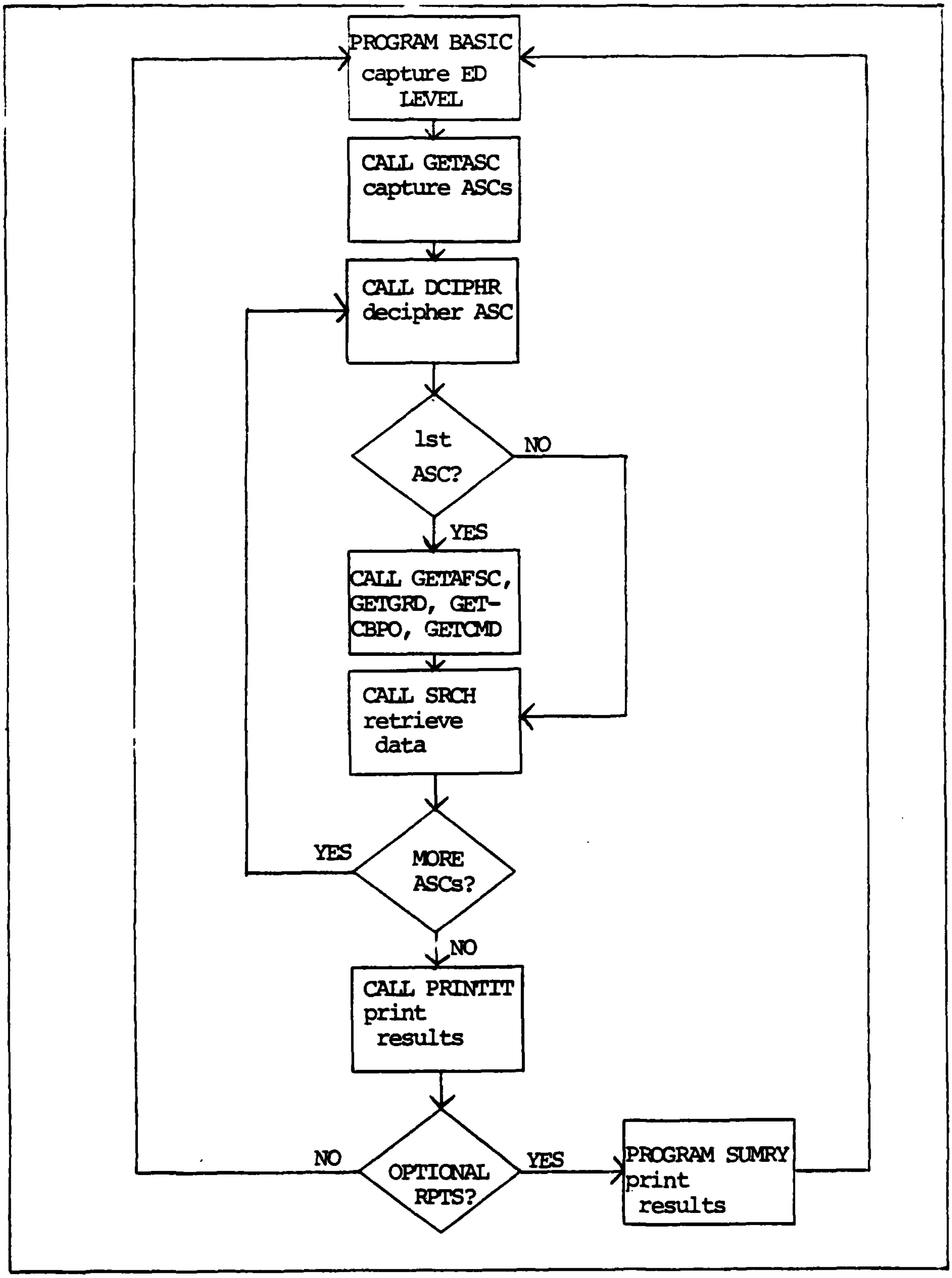


Figure 8, Flowchart of Multiple ASC Capability

Diagram for ADRIS that includes subroutine GETASC is in figure 9.

VERIFICATION - Numerous queries were processed using different ASC combinations. Print lines were inserted into selected subroutines to confirm the initial parameters entered by the user were saved and used again for the remaining multiple ASC entries (when multiple ASCs were input by the user). Finally, queries from the test version of ADRIS were compared against the same queries using single ASC entries in the operational version of ADRIS (since multiple ASCs were not permitted in the original version). Tally comparisons from each system confirmed the output produced was accurate.

4.5 MAJOR COMMAND CODE EXPANSION

BACKGROUND - When ADRIS was developed in 1976-1977, the major command code in the manpower and personnel systems was two characters, as it is today. At that time, the first character of all codes was zero and the second value consisted of numeric values 1-9 and alpha characters A-Z. However, there were only 30 major commands and special operating agencies at the time, so all available characters were not used. The second position was actually a unique character for all codes (no duplication), and therefore Capt Waldron was able to design subroutine GETCMD so that only a single character was input by the user.

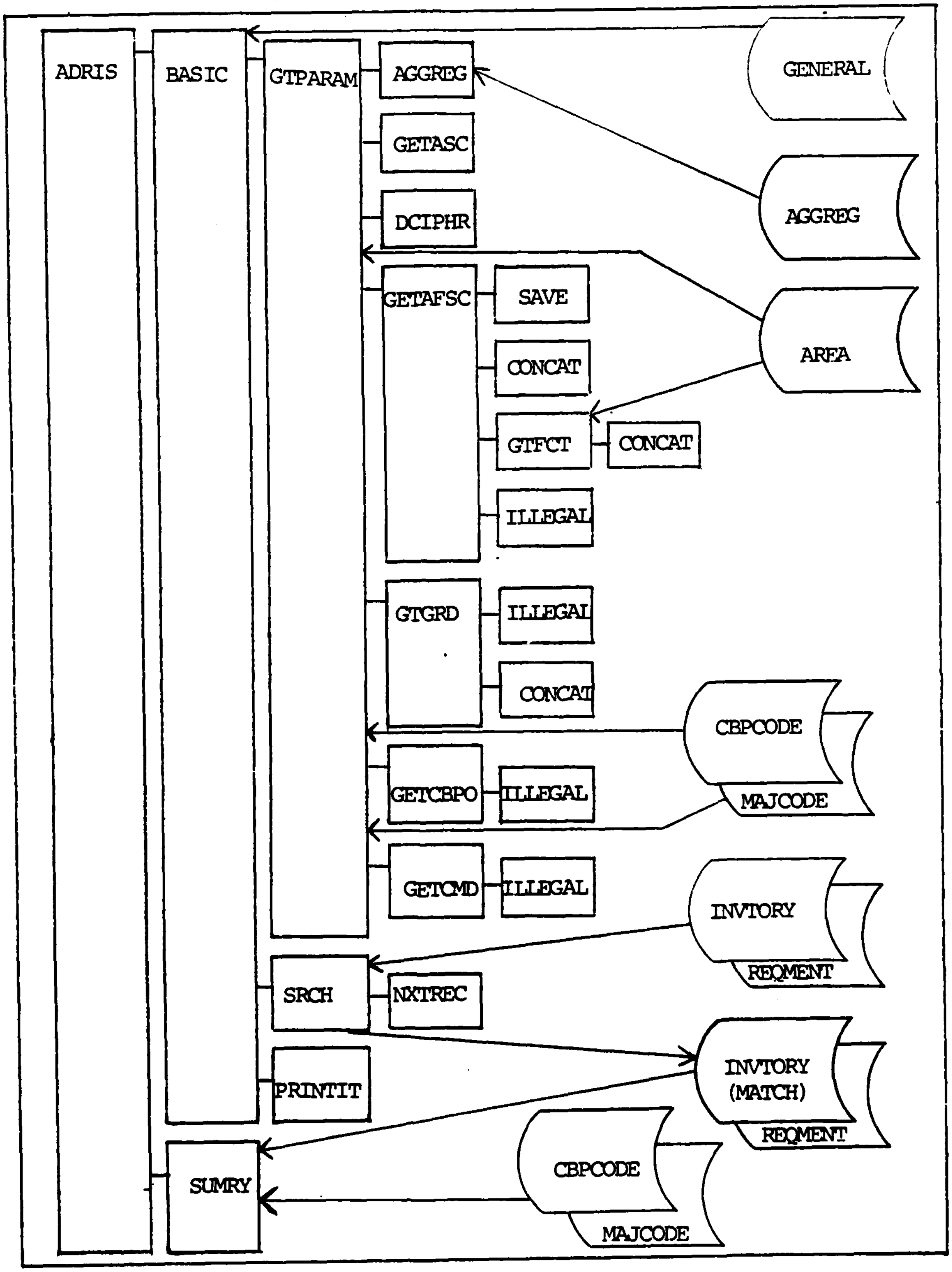


Figure 9, Leighton Diagram of Updated ADRIS Program

This simplified the input for the user, while still providing enough information for an accurate and comprehensive data base search. Since the codes in the Inventory and Requirements data bases were two positions, subroutine GETCMD was designed so a leading zero was added to each single character code input by the user. This resulted in a two position major command code used in the data base search.

ANALYSIS - After a period of systems familiarization, there were serious concerns about the systems capability to handle queries for all major commands because only a single character code was entered by the user. A copy of the current Base Level Personnel System data table for command codes was obtained and there were over 40 in the table.

For many codes (12), the second character was no longer unique. The code for HQ SAC is "0S" and for Hq Space Command it's "1S". When a user specified code "S" in ADRIS, a zero was added and the query automatically searched for records pertaining to Hq SAC. The system would not permit searches for records pertaining to Space Command, as well as the 11 other commands or special operating agencies that contained a second character value that was already in use by another command or special operating agency with a first character value of zero.

This problem with command codes could be even more confusing for a user when the select (*) option was specified, since records pertaining to these 12 commands would be selected when the other selection criteria specified by the user was met. This meant that for queries involving these 12 commands, the output reports would not be consistent depending on the type of query used. It is important to note again that the command code information in the data bases was accurate and comprehensive, but the structure used in ADRIS for entering command codes was incorrect since the number of major commands and special operating agencies increased nearly fifty since Capt Waldron built ADRIS for AFIT use.

DESIGN/IMPLEMENTATION - Subroutine GETCMD was originally designed to accept a single character and it expanded this code into a two position code that was used for the query process. This subroutine was redesigned so that a valid code now consists of two characters instead of one, and the user must enter both characters; if a single character code is entered, an error message is output. Program logic was also deleted that added a leading zero to the code value entered by the user. The user can still enter multiple codes, separated by commas. The user prompt from subroutine GTPARAM pertaining to command code was modified to include a sample input format; this change should reduce user errors when entering codes and will help the new user.

VERIFICATION - Tests were conducted using the old version of ADRIS, the new version requiring a two position code, and the Inventory and Requirements data bases. On the old system, the user could not specify command "1S" since "S" was converted to Hq SAC. However, a select (*) query showed that records for command "1S" were in the data bases. The same query was executed using the new version of ADRIS, and the results for code "1S" agreed with the tallies produced for "1S" using the select all option.

A special operating agency, code "2J" (1947 Admin Support Group), was also tested. Records for it could not be specifically selected using the old system because code "J" was automatically converted to "0J", which is Hq ATC. However, the tests confirmed the modified version of ADRIS corrected this problem. Figure 10 contains sample output from the old and new versions of ADRIS that depicts test results for code "1S".

4.6 DATA FILE REVIEW

BACKGROUND - There are four data files used in the original version of ADRIS. The primary reason for these data files was to permit data changes in ADRIS without making extensive programming changes. Each of these data files has a unique function.

(1)CONVRT: This file contains all obsolete ASCs and the respective replacement ASC for each code that is obsolete. There were 23 obsolete ASCs in the original version of ADRIS.

QUERY USING OLD VERSION OF ADRIS

```
EDLEV= p
ASC=   0cay
AFSC=  c3016
GRADE= 04
CBPO=  ep
MAJCOM=s
```

RESULT

There are no requirements for, or holders of, AD's meeting the criteria you have specified.

QUERY USING OLD VERSION OF ADRIS

Parameters same as used above, except MAJCOM code was changed to * (indicates select against all MAJCOMs)

RESULT

```
LEVEL   ASC    AFSC    GRADE    CBPO      MAJCOM
  P     0CAY   C3016    04       EP         1S
```

Note: This result showed there was a requirement record for "1s" that met criteria specified in the first query. However, code "s" was automatically converted to "0s", and the search did not find a record in Hq SAC meeting the criteria. If a record was found, it would have been erroneous for the user.

QUERY USING NEW VERSION OF ADRIS

Parameters same as used above, except MAJCOM code "1s" was entered instead of "s".

RESULT

```
                 MASTERS
GRADE              REQ           INV
 MAJ                1             0
```

Note: After the standard output, a list all records option was selected, and it showed the following response.

```
MAS    0CAY   C3016     MAJ     PETERSON    SPC CMD
```

This result from the new version confirmed that entering a two position MAJCOM code corrected this problem

Figure 10, Test for Major Command Change

(2)GENERAL: This file contains all ASCs that should remain specific to the fourth position on the Requirements data base and all ASCs that are to be generalized to values 'XY' in the third and fourth position in the Requirements data base. The GENERAL file contained 212 ASCs in the original version of ADRIS.

(3)AGGREG: This file contains eight aggregate codes and the individual ASCs that each code is converted to. The purpose of aggregate coding is to give the user a capability to input a single code and perform a data base search for a series of ASCs that are pre-defined.

(4)AREA: This file contains 21 AFSC codes and the specific AFSCs that each code is converted to. The purpose of the AFSC codes is to provide the user the capability to insert a single code and then perform a data base search for a series of pre-defined AFSCs that are related to a specific career area.

ANALYSIS - Capt Moore said he occasionally received tally information from ADRIS that did not agree with tally reports produced from a system at Washington, D.C. similar to ADRIS. This system is maintained by the Air Force Data Services Center, Washington, D.C. Capt Moore identified the problem as disagreement on the fourth position of the ASC for several codes. As an example, he indicated the system in Washington produced data for ASC "1ASA" (specific to fourth position), but ADRIS would not provide output for ASC "1ASA". Instead, ADRIS provided information for code "1ASY", which includes code "1ASA" but also captures data for additional codes that begin with "1AS".

The GENERAL file was reviewed and code "1ASA" was not found; this proved it was not part of the data base. The solution to Capt Moore's problem was to update the GENERAL file so that "1ASA" was in it, along with any other code changes needed to make the system current. On-line documentation was also needed to briefly explain generalization and provide the user an option for displaying file GENERAL when desired.

There was no one on the AFIT staff responsible for deciding which codes should not be generalized, so after a discussion with Capt Moore, the Data Services Center at Washington D.C. was contacted. Mr John Gates was still assigned there and he was responsible for the system used in Washington that is similar to ADRIS. He provided valuable information to Capt Waldron during the original development of ADRIS, and quickly offered any help needed now to update the data files used in ADRIS so they were again compatible with the system he maintains and with Hq USAF/MPPE policy.

Mr Gates said significant policy changes at Hq USAF/MPPE caused many changes to files in his system corresponding to files GENERAL and CONVRT in ADRIS. After discussing these changes he offered to provide a complete update for all four data files used in ADRIS. These changes were explained to Capt

Moore and he agreed that the data files used in ADRIS should contain the same data used by the system at Washington, D.C.

A thorough analysis of the data furnished by Mr Gates showed that the number of obsolete ASCs in the CONVRT file increased to 30 (originally it was 23) and that the number of ASCs in the GENERAL file was reduced to 100 (originally this was 212). Capt Moore and Dr Bridgman had not used the aggregate and career area options in ADRIS. However, Capt Moore expressed interest in the career area option (explained earlier in this chapter), and therefore these files were also compared with data files maintained by Mr Gates. The AGGREG file in ADRIS was nearly the same, and there were only minor changes to the AREA file used for the career area option in ADRIS. The AREA file was updated to be the same as Mr Gates' file, so any query comparisons performed in the future using career areas codes will be compatible.

DESIGN/IMPLEMENTATION - Minor changes were needed for the AGGREG file and the AREA file. The structure of the AREA file limited the changes to updating the information to this file, with no modifications needed to any ADRIS programs. However, changes to the AGGREG file also required modifications to a format statement used in reading the file. Data changes to the CONVRT and GENERAL files also required coding changes in SPLY and DMND programs that are used to build the ADRIS data bases. These changes were needed primarily due to dimension changes of the data files GENERAL and CONVRT.

In the original source code for programs SPLY and DMND, these dimensions were not highlighted nor did the maintenance manual address them. Extensive documentation has been added to the program so the dimensions can be quickly identified when file size changes occur in the future. The revised maintenance manual also includes information on these dimensions.

User on-line documentation was improved by adding a display option for the GENERAL file if the user desired to see it. This option is available only when the user asks for the system documentation option for ADRIS. This option to display file GENERAL is available only once, prior to the user starting parameter input.

VERIFICATION - After the GENERAL file was updated with changes from Mr Gates, program HASHTST was used to determine if more than two of the ASC's in the new file hashed to the same location. This test was made because three or more ASC's hashing to the same location would cause data base build errors. Before executing the test, program HASHTST was modified to accommodate new file size dimensions for file GENERAL (size was reduced from 212 ASC's to 100). Test results from executing HASHTST showed that only two ASC's hashed to the same location, which is acceptable. Actual program output indicated the following:

```
2 ASC's Hashed to Itable (791)
End HASHTST
```

A similar procedure was used to test the changes in the CONVRT file, which contains obsolete ASC's and their respective replacement codes. A binary tree structure is used to check the ASC on each incoming record with the obsolete codes that have been pre-identified. A new program was developed to build this tree and print the contents of array ITREE which contained all data related to the tree structure. Using the information from the array, a tree was manually constructed (see figure 11) and it showed the new tree structure was in the correct format for operational use. Any ASC searched for in the obsolete list could either be found or a determination made that the ASC was not in the tree in a maximum of five decisions.

Following these tests, the new GENERAL and CONVRT files were used to build a new Requirements test data base, and the CONVRT file was used in building a new Inventory data base. Testing using ASC "1ASA" (a problem code identified by Capt Moore) was conducted using both the test and production versions of ADRIS. The queries and their results are shown in figure 12. The production version used the old file GENERAL which did not contain "1ASA". This caused the code to be generalized to "1ASY" and therefore no results were produced for the query since "1ASA" did not exist in the data base. Using the test version of ADRIS with updated file GENERAL, which contained "1ASA", the query produced output showing that requirements for

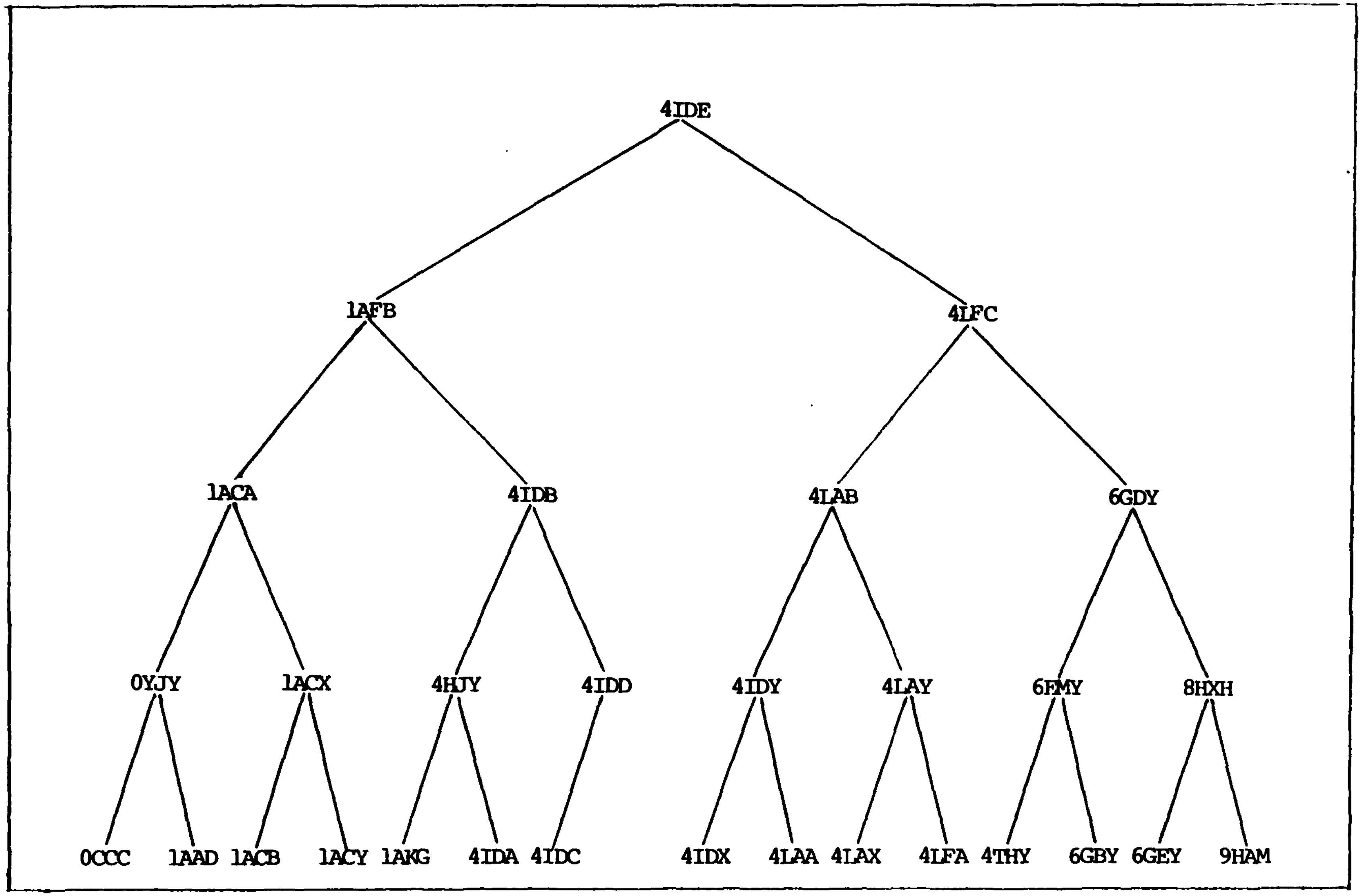


Figure 11, Binary Search Tree Created by Subroutine BLDTREE

QUERY USING OLD VERSION OF ADRIS

```
EDLEV= P
ASC=   1ASA
AFSC=  *
GRADE= *
CBPO=  *
MAJCOM=*
```

RESULT

There are no requirements for, or holders of, AD's meeting the criteria you have specified.

QUERY USING NEW VERSION OF ADRIS

(same parameters as entered above)

RESULT

```
                       MASTERS
GRADE                    REQ            INV
  02                      4              0
  03                     30              0
  04                      8              0
  05                      2              0
  06                      1              0
```

(This result confirmed ASC was in requirements tape furnished by Hq AFMPC. By changing the GENERAL file, this ASC was not generalized; the fourth character was not changed)

Figure 12, Test for Change in ASC Generalization

ASC "1ASA" did exist.

This requirements information is important to AFIT planners, so ASC generalization performed by ADRIS must be consistent with guidelines established by Hq USAF/MPPE. Additional testing on other codes newly added to file GENERAL or deleted from it confirmed that the test version of ADRIS was handling generalization correctly.

4.7 REPORT MODIFICATIONS

BACKGROUND - There are seven report formats available from ADRIS. One format is standard for every query, and then there are six optional reports available. Tally information contained in the reports is displayed by using various combinations of ASC, AFSC, grade, education level, CBPO and major command. The primary emphasis in this enhancement was to make reports more readable for ADRIS users as well as for educational planners and AFIT managers who might review the ADRIS output. Grade, base, major command, and education level were the items identified by Capt Moore as the most important for translated output. The PRINTIT subroutine produces the standard report format and the SUMRY overlay produced the six optional reports.

ANALYSIS - Program efficiency was important in this modification since there would be some duplication involved in producing more readable output for seven reports that contained similar data. Another important consideration concerned the nature of the data. Education level and grade data were fixed; there are four available education levels used in ADRIS and there are six officer grades, excluding General officers, in use by the Air Force. The codes and meanings for these codes are not likely to change.

However, data pertaining to major commands and bases does change. New bases or major commands will start operating and some bases and major major commands can be expected to close. Although these changes do not occur often, they do happen. For example, Capt Waldron allowed for 36 major commands when he coded ADRIS for AFIT use and there were only 30 major commands and special operating agencies at that time. In 1983, there were over 40 codes. Since changes in base and major command data can be expected, a more flexible means on handling these was needed.

DESIGN/IMPLEMENTATION - Grade and education level values were stored in subroutine PRINTIT and overlay SUMRY, since only four and six values were involved, respectively, that were not likely to change. An array was set up for each item and array values were loaded using type DATA. Data for bases and major commands was stored in new files, CBPCODE and MAJCODE. Data files were used because they simplify the update process when changes are necessary. Another design concept involved building in back-up print capability so that the addition or deletion of major command or base codes in the files furnished by AFMPC would not cause the reports to abort.

This was done by designing a back-up print capability so any new data codes furnished by AFMPC would be used whenever the data files MAJCODE or CBPCODE did not contain the code and its cleartext name. This concept served two purposes. First, the user still received a valid report even if a base or major command code was not recognized by the system. Second, the appearance of code data in the reports signaled that maintenance would be needed to make the data table(s) current. These tables were also designed so they could be used to provide on-line documentation to ADRIS users who needed immediate access to available codes and major command and base names.

VERIFICATION - The reports modification did not add new query logic to the ADRIS. This modification only changed the appearance of the standard output report and six optional reports. After the reports modifications were made, testing was accomplished by executing many queries with different combinations of grades, MAJCOMs, CBPOs, ASCs, education levels and report options. These queries were run using the production version of ADRIS and then the same queries were input using the test version. Each query was compared item for item and the results confirmed that while the appearance of the reports was significantly different from one system to the other, the data content of each pair of reports was exactly the same.

4.8 DATA BASE ANALYSIS

BACKGROUND - The data bases were originally designed to reduce search time by placing all ASCs with the same first numeric value (0,1,2,etc.) together. A pointer file is built during data base generation that is used during the search process. This pointer file makes the search process very efficient. The records size is small, containing 17 characters each.

The data base design seems well tailored for the system and the queries it handles. The frequency for data base generation is every three months and the amount of system resource time needed to accomplish this is very minimal. Data base storage for both files on the CDC CYBER totals only 33 blocks, even following the addition of additional records for the Bachelor's Degree modification. Therefore no changes were considered for the structure of both the Inventory and Requirements files.

4.9 PROGRAM DOCUMENTATION

Each program and subroutine contained a general description of the activity performed. Significant data names were also documented in programs and some subroutines. However, documentation explaining the activity in specific areas of the program was very sparse.

The modifications made to ADRIS involved most of the programs and subroutines. To properly analyze the system and create the best design for the major modifications, extensive analysis of much of the existing code in the system was required. This resulted in a good understanding of much of the program logic in the system. Approximately 250 lines of comments were added to the programs and subroutines to better explain what they do and how they do it. While many of these comments pertain to new lines of code that were added, a lot of the comments were inserted to explain existing areas of code.

These additional comment lines should greatly simplify any future maintenance or enhancements performed on ADRIS.

Chapter 5

MODIFICATION SUMMARY AND VALIDATION

This chapter summarizes significant modifications made to ADRIS and identifies changes that enhance system efficiency. Validation of the system modifications with users participating in testing is also discussed.

The original thesis topic was to make ADRIS easier for its users to operate and understand. However, during the requirements analysis it became clear there were serious problems in the system (unknown to users) unrelated to user friendliness. These problems needed attention, and two concerns were foremost.

First, some queries were producing erroneous results because the data tables were not correct. Over six years went by since Capt Waldron completed his thesis work and there was no indication nor any record showing that any maintenance was performed on the system during this period.

Secondly, there was information in the data bases that could not be reached with some queries because of an error in the majoı command code structure and discrepancies in generalizing some ASCs. These problems needed to be corrected so tally information generated from ADRIS would be accurate to support educational planning. Therefore, these problems took on a higher priority than changes related to user friendliness.

Correct information for all existing data files was obtained from the Data Services Center and then each file was updated. Extensive research was performed on major command codes, and then the code structure was modified in ADRIS so records pertaining to all commands could be retrieved. The career area option was corrected so it worked properly, and a hidden limitation on the number of CBPO code entries was corrected.

Another unique problem involved the program library. In its original form, the library was structured for the UPDATE utility on the CDC CYBER 74 computer. While this utility was useful for Capt Waldron's work, it was not effective for the ADRIS modifications. EDITOR was chosen for text editing, and the program library had to be re- structured to support its use. Data files CONVRT and GENERAL were reconfigured to reduce line characters from 80 to 72, so their size would be compatible with the EDITOR. These file size changes caused minor modification to several READ formats, but the overall concept of reducing the characters per line made the data files easier to work with.

A totally new concept for ADRIS was translation of coded data on all output reports. Two new data files were created, one containing CBPO

information and the other majo: command information. These files were structured so they could also serve as user documentation in the program. Two other existing data files, CONVRT and GENERAL, were also included in on-line documentation. By using these files for a dual purpose, data redundancy was avoided. Using these operational files for documentation also provides a user with the exact information used in the system; when data in the file is changed, the user will have access to current documentation without waiting for an updated users guide.

5.1 IMPROVING ADRIS EFFICIENCY

The following enhancements were designed to increase system efficiency:

(1)A user can terminate on-line documentation after any query. Turning it off reduces time spent at the display screen and allows the system to react faster as new parameter information is input.

(?)Establishing the multiple ASC input capability allows the user to enter up to ten specialty codes. Prior to this modification, parameters for education level, AFSC, grade, CBPO code and major command had to be entered for each ASC, even when the parameters did not change. When multiple ASCs are entered, the system was also redesigned so that tally information would be accumulated for all the ASCs. After data base searching is complete, the user gets a report that is summarized. In the old version of ADRIS, a user manually totaled this information from each query involving a different ASC. The time savings for the user from these changes is significant.

(3)The use of data files for two purposes was discussed earlier in this chapter. Data redundancy is avoided and the user has accurate on-line documentation immediately after a data file change is made.

(4)There were references to academic education level "2" through- out the program. Program logic keyed on education level "2" in several places, but this code was not needed since data tapes from Hq AFMPC did not contain it.

(5)File redundancy in the program library was eliminated. This change saves file space and also makes the program library simpler to use.

5.2 VALIDATION

Each modification was verified using multiple test queries, and in many instances duplicate queries were run using the operational system to make sure data base searches were accurate. Tallies from the test and operational systems were always compared to insure accuracy in each modification.

The test system underwent further validation by the primary users, Capt Moore and Dr Bridgman. On 3 October 1983 the version of the retrieval system containing all changes except the capability to retrieve Bachelor's Degree records was made available to them. Their testing efforts resulted in several additional enhancements to aid the user and also in uncovering an error related to the number of CBPO codes allowed. On 24 October 1983 a newer version of ADRIS was catalogued to the test library that contained retrieval capability for Bachelor's Degree records.

Queries run by Capt Moore and Dr Bridgman, coupled with the extensive systems test performed during the modification to ADRIS, confirmed that tally reports produced by the new system were accurate.

Chapter 6

CONCLUSION

The modified version of ADRIS is operational and available to anyone who previously had access to the system. The version developed by Capt Waldron was being deactivated and replaced by the new version. A copy of all ADRIS programs, data files and procedures was turned over to Dr Bridgman and to AFIT/ADO.

The following recommendations were made concerning future modifications to ADRIS and on-going system maintenance:

(1)An off-line report capability should be developed. This would allow the user to enter queries without waiting for tally reports to appear on the CRT screen.

(2)AFIT/ADO should assume a more active role in performing routine systems maintenance on ADRIS. Dr Bridgman has been doing the quarterly data base updates, but support from AFIT/ADO is needed to accomplish table updates and other routine maintenance on ADRIS. As new CBPOs and major commands become operational or existing ones deactivate, the CBPCODE and MAJCODE files will need to be changed. When more ASCs become obsolete or there are changes to ASC generalization policies at Hq

USAF/MPPE, the CONVRT and GENERAL files will require changes.

(3)Mr Gates at the Air Force Data Services Center should be contacted periodically to obtain updates on ASC conversion and generalization policy. He can provide changes when they occur as a result of Hq USAF/MPPE policy changes.

ADRIS tally reports over the last six years became increasingly inaccurate because the system was not properly maintained. Reports are now correct because of the updates and changes performed on the system. However, report accuracy will decline again in the future if the system is not properly maintained.

Bibliography

AFR 35-25.Educational Classification and Coding Procedures, Hq USAF/MPPE, Washington,D.C., 13 December 1982.

AFR 36-19.Advanced Academic Degree (AAD) Management System, Hq USAF/MPPE, Washington,D.C., 22 March 1983.

CYBER Loader Version 1 Reference Manual. Control Data Corporation, Califormia, 1979.

INTERCOM Guide. Revision A, September 1976, Air Force Institute of Technology, Wright-Patterson AFB, Ohio.)

Organick, Elliot I. and Meissner, Loren P.FORTRAN IV. Phillipines: Addison-Wesley Publishing, 1974.

Peters, Lawrence J.Software Design: Methods and Techniques. New York: Yourdan Press, 1981.

Appendix A

GLOSSARY OF ACRONYMS

AAD	Advanced Academic Degree
AADMS	Advanced Academic Degree Management System
ADRIS	Advanced Degree Requirements Information System
AFERB	Air Force Education Requirements Board
AFIT	Air Force Institute of Technology
AFMPC	Air Force Manpower and Personnel Center
AFSC	Air Force Specialty Code
ASC	Academic Specialty Code
AUTODIN	Automated Digital Network
CBPO	Consolidated Base Personnel Office
CDC	Control Data Corporation
CYBER 74	CDC Computer
DAR	Data Automation Requirement
EDITOR	Text Editor for CDC CYBER Computer
INTERCOM	Interactive Terminal System
MAJCOM	Major Command
MPPE	Hq USAF Office Responsible for Education Programs
NOS	Network Operating System
NOS/BE	Network Operating System/Batch Environment

UPDATE Utility for Handling Programs on CDC CYBER Computer

Appendix B

USER'S GUIDE

B.1 Purpose of ADRIS

This user's guide is an updated version of the original guide developed for ADRIS by Capt Matthew Waldron.

The purpose of ADRIS is to use the speed of an interactive computer program to provide detailed and summary information about the inventory of Air Force officers possessing Advanced Academic Degrees (AAD) in any Academic Specialty Code (ASC) (and Bachelor's Degrees in specific ASCs) and the job positions that require AAD officers. Table III identifies those ASCs associated with Bachelor's Degree records.

Table III, ASCs Associated with Bachelor's Degree Records

CODE	ACADEMIC AREA
0YYY	Computer Tech, Operations Research, Misc
4YYY	Engineering
6YYY	Mathematics
8YYY	Meteorology, Physics

The Inventory and Requirements information is contained in two data bases built from magnetic tapes furnished quarterly by the Air Force Manpower and Personnel Center (AFMPC). The two tapes are extracts from Uniform Officer Record and Manpower Authorization files maintained at Randolph AFB, Texas. The Inventory data base contains the education level, ASC, Air Force Specialty Code (AFSC), grade, base, and major command for each AAD and selected Bachelor's Degree officers, and the Requirements data base contains the same information for each AAD position.

The main product of ADRIS is an Inventory and Requirements count of officers and positions satisfying the criteria selected by the ADRIS user (when a query for Bachelor's Degree is run, the product will be only an Inventory count of the officers satisfying the selection criteria). The criteria consist of values chosen by the ADRIS user for the six parameters: education level, ASC, AFSC, grade, base, and major command. The ADRIS user can optionally obtain more detailed summaries of the data base information matching the selection criteria. Summaries by ASC, AFSC, base, and command may be printed, and all records selected during the query can also be printed.

B.2 USING ADRIS

The ADRIS program and data bases reside on the Control Data Corporation CYBER (CDC) 74 computer at Wright-Patterson AFB, Ohio. ADRIS is accessible anytime the interactive terminal (INTERCOM) system is operating. Normal operation hours are 0830 to 2400 hours, Monday through Saturday, and 0830 to 2200 hours on Sunday.

Special knowledge about computer systems is not needed to run ADRIS. The program provides instructions to the user and performs error checks on the query selection parameters input by the user. A brief summary of terminal operation instructions is contained in Section B.3.

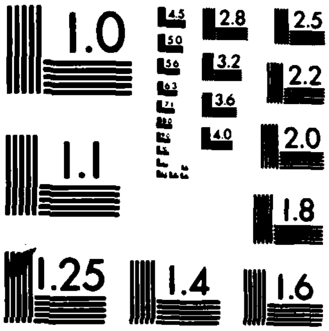
LOGIN AND STARTING THE PROGRAM

The ADRIS user must first login to the INTERCOM system. Users unfamiliar with terminal procedures should now read Section B.3. The ADRIS problem number and account number is T770008. The user must have a valid account number on the CYBER 74 computer, and the ADRIS program can be executed from any active account number. The login line is as follows:

```
login,(account number),(password for the account number)
```

Once the login is complete, the system will respond with:

AD-A138 167 MODIFICATION TO THE ADVANCED DEGREE REQUIREMENTS 2/2
INFORMATION SYSTEM(U) AIR FORCE INST OF TECH
WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING
UNCLASSIFIED G P RANALLO DEC 83 AFIT/GCS/MA/83D-7 F/G 5/2 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

```
COMMAND-
```

The user can then activate the ADRIS program by entering:

```
get,(codeword)/un=t840111
```

This codeword may be obtained from the AFIT School of Engineering Director of Academic Programs and Operations. After the attach is complete, the user then enters the following command:

```
begin,afit,(codeword)
```

ADRIS will now be ready to run under the account number that the user originally logged onto the system.

INTERACTING WITH THE PROGRAM

Program-user interaction is largely self-explanatory with the on-line documentation providing necessary instructions and then printing an equals sign (=) followed by a pause when the user must enter a response. Users must not insert blanks between any parameters entered or immediately after the (=) sign.

Once the user has typed a response the RETURN or CR key must be depressed to transmit the response. If a syntactical error is detected ADRIS will provide an error message and direct the user to reenter the data. Logical or miskeyed errors cannot be retracted once the RETURN key has been struck. If this occurs, the user must wait until the data base search is complete and the program has recycled back to the point where the error was made (another option is to stop ADRIS and start over, which is explained later).

If the user detects an error before the line has been transmitted, the error may be corrected by depressing the CTRL key on the terminal and then pressing H to backspace to a point where the entry can be corrected by typing over the faulty letter(s). The error can also be corrected by using the backspace key to return to the point of the error, and then retyping the line again from that point.

The initial program request is for the user to identify as a new or old user. When the user indicates that he/she is "Familiar with ADRIS", abbreviated instructions and prompts are provided so the parameters can be entered quickly. Users who indicate they are not familiar will receive extensive documentation that provides examples for each required parameter. A user who chooses to receive extensive documentation can stop the documentation after their first query is finished, or at the conclusion of any subsequent query.

STOPPING ADRIS

The program may be terminated during printing of output by pressing the ESC key, followed by the % key and then the A key. ADRIS can be stopped during a pause (for example, waiting for user input) by entering %A. ADRIS can be restarted by entering AADMS after the terminal aborts the program and COMMAND- appears on the display screen. Normal program termination is directed by the user entering "yes" at the end of a query when asked if "Done". If "no" is entered, the user may continuing entering parameters to run more queries. If "yes" is entered, ADRIS operations will terminate and the user can either disconnect from the CYBER 74 computer by entering "logout" or continue working with the CYBER 74 for other systems work.

SEARCH CRITERIA

To execute a query, the user must enter values for the six search parameters. The data bases are then searched to find Inventory and Requirements entries which the the selection parameters. When a single ASC is entered, a tally of the results is then printed on the display screen as well as the ratio of Inventory records selected to Requirements records selected. When multiple ASCs (up to 10) are entered, a single tally for the results of all ASCs is printed.

The parameters and rules governing the entry of their values follow. When the user wants to select all possible values for a specific parameter, an asterisk (*) should be entered.

(1)EDUCATION LEVEL - Enter "p" for Master's Degree, "q" for Master's Degree plus, "3" for PHD students, "r" for PHD or "n" for Bachelor's Degree. Queries using "p" will select records where level is P, Q, or 3. Queries using "r" will only select records where level is R. Queries using "n" will select records where level is N or O (Bachelor's Degree plus). The select (*) option will search for education levels P, Q, R, and 3; Bachelor's Degree records are not included in a select (*) search.

(2)ASC - The user must enter a single ASC or multiple ASCs (up to ten, each separated by a comma) as identified in AFM 300-4, ADE AC-030, or a single aggregate ASC can be entered as identified in table IV. A "y" in a character position of the ASC denotes no academic specialization for that component of the ASC. If the entry contains a "y" the user will be asked to specify whether he/she means the specific ASC input or all ASCs with any allowable character in the "y" position(s). Most Requirement ASCs are specific only to the first three characters, because of the ASC generalization policy explained

Table IV, Aggregate Codes for ASC

AG CODE	ASCs	DESCRIPTION
AAAY	4AYY, 4BYY, 4EYY, 4KYY, 4MYY	Aeronautical-Astronautical and Mechanical Engineering
AABY	6YYY, 8CYY, 8HYY	Basic Sciences
AACY	0CBY, 0CCC, 6GGY, 6GJY, 6IYY	Data Reduction and Analysis
AADY	4BCY, 4EDE, 4IHY, 4MHB, 4TAY, 6EFY	Guidance and Control
AAEY	4ICE, 4ICF, 8HFH, 8HOY	Solid State
AAFY	0YKY, 1AMG, 1ASY, 4ACF, 4BDY, 4LDC, 4TYY	Systems Management
AAGY	1AGY, 1APY, 1ASY	Program Management
AAHY	0YEY, 1AKG, 4LCF, 4LFY, 4TGY, 4THY, 4TIY, 4TKY, 6EHY, 6EMY, 6EOY, 6GLY, 8HXH	Quantitative Analyses

Table V, ASC Exceptions to Generalization Policy

Codes Specific to Four Char				Codes Changed to YY (3rd/4th)			
0YLA	0YLB	0YLC	0YLD	1AXY	2AXY	2BXY	2CXY
0YLE	0YLF	0YLG	0YLH	2FXY	2GXY	2HXY	2IXY
0YLI	0YLJ	0YLK	0YLL	3AXY	3BXY	4AXY	4BXY
0YLM	0YTA	0YTB	1AGA	4CXY	4DXY	4EXY	4FXY
1AMH	1AM1	1AMJ	1AMM	4GXY	4HXY	4IXY	4KXY
1AMS	1ASA	2CAB	4KAA	4LXY	4MXY	4NXY	4OXY
4KAB	4KAC	4KAD	4LFA	4PXY	4QXY	4RXY	4TXY
4LFB	4LFC	8HMJ	8HXA	6AXY	6BXY	6CXY	6DXY
8HXB	8HXC	8HXD	8HXE	6EXY	6FXY	6GXY	6HXY
8HXF	8HXG	8HXH	8HXI	6IXY	6JXY	8AXY	8BXY
8HXJ	8HXK	8HXL	9HAM	8CXA	8CXX	8CXY	8DXY
9HAN				8EXY	8FXY	8GXY	9AXY
				9BXY	9CXY	9DXY	9EXY
				9FXY	9BXY	9HXY	

in the thesis. Table V contains a list of exceptions to this policy (this list is also available over the display screen). Table VI contains a list of obsolete ASCs and their replacement codes. The select (*) option can also be used, and will search records containing any ASC. However this is a longer search process and should be used cautiously. When multiple ASCs are entered, the user must be careful not to duplicate an ASC by specifying it again or the reports will not be correct. For example, if "0cay" is entered and a "2" is input to request information on this ASC and all subspecialties, the subspecialties will include "0cab". If "0cab" is also entered as a specific ASC, the results for this code will be included twice in the report totals.

(3)AFSC - A single value or multiple values separated by commas are permitted. Ranges of values can also be used, such as 26xx-29xx or 513x-514x, where the "x" shows the digits over which the range extends. Career area descriptors for common AFSC groups may be entered as shown in table VII, and career area information is also available on the display screen when using ADRIS. Any combination of these AFSC entries is allowed, as long as the entry fits on one line.

(4)Grade - A single grade or multiple grades separated by commas must be entered. A number between 1 and 6 is used to represent grades second lieutenant through colonel, respectively. A zero or letter O can also be placed in front of

Table VI, Obsolete and Replacement ASCs

OLD	NEW
0CCC	0YEY
0YJY	0YEY
1AAD	1AAB
1ACA	0CBY
1ACB	0CAC
1ACX	0CAD
1ACY	0CAB
1AFB	1AFD
1AKG	0YEY
4HJY	4HYY
4IDA	0CBA
4IDB	0CBB
4IDC	0CBC
4IDD	0CBD
4IDE	0CBE
4IDX	0CBX
4IDY	0CBY
4LAA	0CCA
4LAB	0CCB
4LAX	0CCX
4LAY	0CCY
4LFA	0YEY
4LFC	0YEY
4THY	0YEY
6EMY	0YEY
6GBY	0CDA
6GDY	0CDB
6GEY	0CDC
8HXH	0YEY
9HAM	0YMY

Table VII, Career Area Codes

CODE	AFSCs	CAREER AREA
ADMI	70XX	Administration
CHAP	89XX	Chaplain
CIVI	55XX, 62XX	Civil Engineering and Services
COMM	30XX	Communications and Electronics
COMT	005X, 67XX, 69XX	
EDUC	0900, 0940, 0950 75XX	Education and Training
HIST	0930	Historian
INFO	79XX	Information
INTE	0910, 57XX, 80XX	Intelligence
LAWY	88XX	Law
LOGI	0005, 004X, 009X 31XX, 40XX, 46XX 60XX, 63XX, 64XX 65XX, 66XX	Logistics
MANP	74XX	Manpower
OPER	002X, 003X, 006X 007X, 008X, 021X 051X, 10XX, 11XX 12XX, 13XX, 14XX 15XX, 16XX, 17XX 18XX, 21XX, 22XX 23XX, 52XX	Operations
PERS	001X, 0920, 73XX	Personnel
SCIE	26XX, 27XX, 28XX	Scientific and Devlop Engineering
SECU	81XX	Security Police
SPEC	82XX	Special Investigations
WEAT	25XX	Weather
COMP	0960, 51XX	Computer Technology
OPRE	2691, 2695	Operations Research
PIPE	0001, 0003 0004 0006, 0007, 0008 0101, 0102, 0103 0104, 0105, 0106 0110, 0111, 0112	Pipeline

the grade(s) selected. General officers are not included in the data bases and therefore a select cannot be made for these grades. Colonels are included for information only since there are not any quotas for AFIT education for this grade (O6). If select (*) is used, the query will consider records with any grade.

(5)CBPO - The user must enter a single or multiple (separated by commas) two character code as defined in table VIII. On-line documentation is also available that provides the two character code for each CBPO. If select (*) is used, all CBPOs will be considered during the data base search.

(6)MAJCOM - The user must enter a single or multiple (separated by commas) two character code as defined in table IX. On-line documentation is also available that provides the two character code for each MAJCOM. If select (*) is used, all MAJCOMs will be considered during the data base search.

EXAMPLE:

```
EDLEV=p
ASC=0cyy
ENTER 1 TO DESIGNATE ONLY THIS SPECIFIC ASC
      2 TO SUMMARIZE THIS ASC + ALL ITS SUBSPECIALTIES
=2
AFSC=51xx
GRADE=03,04
CBPO=*
MAJCOM=0s
```

This query would result in the Inventory and Requirements status of a captains and majors assigned to Strategic Air Command with a 51xx AFS and a Master's Degree in computer technology. All ASCs beginning wit "0c" would be considered in this search. The following standard repo would be printed on the display screen:

Table VIII, CBPO Codes

CBPO	CODE	CBPO	CODE
AFIT	WY	Alconbury	AH
Altus	AM	Anderson	AT
Andrews	AU	ARPC(RES)	S7
Aviano	AY	Barksdale	BB
Beale	BO	Bentwaters	BF
Bergstrom	BH	Bitburg	BL
Blytheville	BN	Bolling	BP,WG
Brooks	BV	Cp New Amstrdam	CC
Cannon	CD	Carswell	CF
Castle	CH	Chanute	CK
Charleston	CL	Clark	CP
Columbus	CO	Davis-Monthan	DF
Dover	DM	Dyess	DW
Edwards	EB	Eglin	ED
Eielson	EH	Ellsworth	EJ
Elmendorf	EL	England	EM
Fairchild	FC	FE Warren	FW
Ft Meade	FT	George	GB
Goodfellow	GF	Grandfork	GM
Greenham Comm	GC	Griffiss	GW
Grissom	BX	Hahn	HB
Hancock	HF	Hanscom	LK
Hellenekon	AX,IK	Hickham	HL
Hill	HP	Holloman	HS
Homestead	HV	Howard	AF
Hurlburt	EE	Incirlik	IN
Kadena	KB	Keflavik	IC
Keesler	KF	Kelly	KH
Kirtland	KV	KI Sawyer	KY
Kunsan	KU	Lajes	LC
Lackland	LA,LB,ZB	Lakenheath	LD
Langley	LE	Laughlin	LJ
Lindsey	WU	Little Rock	LP
Loring	LS	Los Angeles	LU
Lowry	LL,LW	Luke	LY
Macdill	MA	Malmstrom	MB
March	MD	Mather	ME
Maxwell	MG	McChord	MH
McClellan	MU	McConnell	MK
McGuire	MN	Mildenhall	ML

Minot	MP	Misawa	MO
Moody	MT	Mountain Home	MW
Myrtle Beach	MY	Nellis	NJ
Norton	NV	Offutt	OD
Osan	OP	Pease	PJ
Pentagon	HH	Peterson	EP
Plattsburgh	PS	Pope	PV
Presidio	PM	Patrick	AK,PF
Ramstein	RF	Randolph	RJ
Reese	RM	Rhein-Mein	RP
Robins	RX	San Vito	SB
Scott	SF	Sembach	SJ
Seymour-Johnson	SM	Shaw	SP
Sheppard	SQ	Spangdahlem	ST
Stuttgart	PE	Sunnyvale	SX
Tempelhof	BU	Tinker	TE
Torrejon	TJ	Travis	TP
Tyndall	TX	Upper Heyford	UP
USAFA	US,ZE	Vance	VH
Vandenberg	VQ	Wheeler	WR
Whiteman	WT	Williams	WV
Wright-Patt	WE	Wurtsmith	WZ
Yokota	YM	Zaragoza	ZG
Zweibrucken	ZN		

Table IX, Major Command Codes

COMMAND	ABBRV	CODE
Aerospace Defense Center	ADZ	0Z
AF Acct/Finance Center	AFAFC	0E
AF Audit Agency	AFAA	06
AF Communications Command	AFCC	0Y
AF Commissary Service	AFCOMS	1X
AF Combat Operations Staff	AFCOS	2H
AF Elements	ELEMNTS	3V
AF Elements Europe	ELM EUR	3G
AF Engring/Services Center	AFESC	1W
AF Intelligence Service	AFIS	05
AF Inspection/Safety Center	AFISC	02
AF Logistics Command	AFLC	0F
AF Legal Services Center	AFLSC	2E
AF Manpower/Personnel Center	AFMPC	09
AF Medical Services Center	AFMSC	2F
AF Office of Security Police	AFOSP	08
AF Reserve	AFRES	0M
AF Review Board	REV BRD	2M
AF Systems Command	AFSC	0H
AF Service Info/News Center	AFSINC	2G
AF Simpson Hist Research Center	AFSHRC	2K
AF Tech Applications Center	AFTAC	2L
AF Test/Evaluation Center	AFTEC	03
Air National Guard	ANG	34
Air National Guard Support	ANG SPC	2I
Air Reserve Personnel Center	ARPC	0I
Air Training Command	ATC	0J
Air University	AIR UNV	0K
Alaskan Air Command	AAC	0A
Defense Mapping Agency	DMA	88
Electronic Security Command	ESC	0U
Headquarters USAF	HQ USAF	0N
Military Airlift Command	MAC	0Q
Military Traffic Management	MIL TMO	77
Office Special Investigations	OSI	07
Pacific Air Forces	PACAF	0R
Reserve Central Management	RES MGT	3I
Strategic Air Command	SAC	0S
Space Command	SPC CMD	1S
Tactical Air Command	TAC	0T
US Air Force Academy	USAFA	0B
US Air Forces Europe	USAFE	0D
1947 Admin Support Group	1947 GP	2J

	MASTERS	
GRADE	REQ	INV
CPT	21	13
MAJ	11	13
TOTALS	32	26
INV/REQ		.8

(NOTE: TOTALS EXCLUDE COL'S

SUMMARIES

The user may request optional summary reports, based on the criteria already entered for a query. There are six optional reports available, and the following prompt is provided to the user to help in selecting these options:

```
ENTER ONE NUMBER CORRESPONDING TO THE TYPE OF REPORT YOU
DESIRE (DO NOT ENTER AN ASTERISK)
          1. ACADEMIC SPECIALTY CODE
          2. AFSC
          3. CBPO
          4. MAJOR COMMAND
          5. SPECIAL MAJCOM SUMMARY (N/A FOR EDLEV = N)
          6. LIST ALL RECORDS
=
```

The AFSC summary prints each different AFSC and the tally by grade for Inventory and Requirements. The Special MAJCOM summary prints ASCs, for each command, by grade, for Inventory and Requirements (no output is furnished if there are not any requirements for the parameters used in the query.

The AFSC, ASC, and Special MAJCOM summaries require the user to indicate the degree of character specificity. For example, assume that the original ASC parameter value was "Ocyy", with all subspecialties requested. Then an ASC summary with degree of specificity of 3 would result in a report with tallies for Ocay, Ocby, etc. An ASC summary with degree of specificity 4 would result in a report with tallies for Ocaa, Ocab,....Ocba, Ocbb,.....

When optional report 6 is chosen, all records selected during the search will be printed. However, no summary data is furnished with this report.

B.3 TERMINAL OPERATION INSTRUCTIONS

ADRIS can be accessed by any terminal connected to the CYBER 74 computer system. Most terminals used to access the CYBER 74 computer are connected through the GANDALF switching network, and the following procedures are applicable to this network. Terminals are available in room 133, AFIT School of Engineering, building 640, and there are individual terminals in many offices in the School of Engineering and at AFIT Headquarters, building 125.

The power switch to the terminal must be on, and it is normally located at the lower right position. When the power is on, the user must make sure the baud rate (speed of data over the communication lines) is at the proper setting. The CYBER 74 operates at 300 and 1200 baud, and ADRIS will execute using either of these baud rates. Since 1200 baud is much faster, it is the best choice when using ADRIS. Each terminal usually will have instructions nearby explaining how to set the baud and connect to the different computer systems available at AFIT.

To use the VISUAL 100 terminals in room 133 at the School of Engineering, the user depresses the set up key followed by numeric key 5. Then the baud rate can be set by depressing numeric keys 7 and 8, which will cause two sets of number to change at the lower right corner of the terminal display screen. When both numbers are at 300 or 1200, the user can press the set-up key again. Now the terminal is ready to be connected to the CYBER 74 computer.

The GANDALF switch (blue box adjacent to the terminal) must now be set to either 40 (for 300 baud) or 41 (for 1200) baud. This should be done with the ready switch down. Flipping the ready switch up after the number is properly set will connect the terminal to the CYBER 74, provided a port is available. If the ready light remains on, the user should press the RETURN key and then the login procedure can start. When a connection cannot be made because all ports are busy, the user can queue (wait) for the CYBER 74 by setting the number dial to 00 (instead of 40 or 41). This will cause the ready light to stay on, and then by pressing RETURN the user will be directed to

ENTER CLASS (40 or 41 is entered at this time using the terminal). By entering "y" when asked if queuing is desired, the user will go into the wait queue and will be advised over the terminal when to login. Normally there are detailed instructions on the GANDALF switches to explain most of these procedures. Once the user is connected to the CYBER 74, the instructions for attaching and executing ADRIS found earlier in this chapter can be used.

Appendix C

MAINTENANCE GUIDE

This original version of this guide was created by Capt Matthew Waldron for ADRIS users. This version was updated from the original version. It contains instructions for building new Inventory and Requirements data bases, data file structure, a file directory and programming notes.

C.1 Building New Data Bases

This section is a procedural guide to be followed to build new data bases from magnetic tapes furnished by Hq AFMPC. If any problems are encountered or programming changes desired, assistance should be requested from the AFIT/ADO ADRIS monitor.

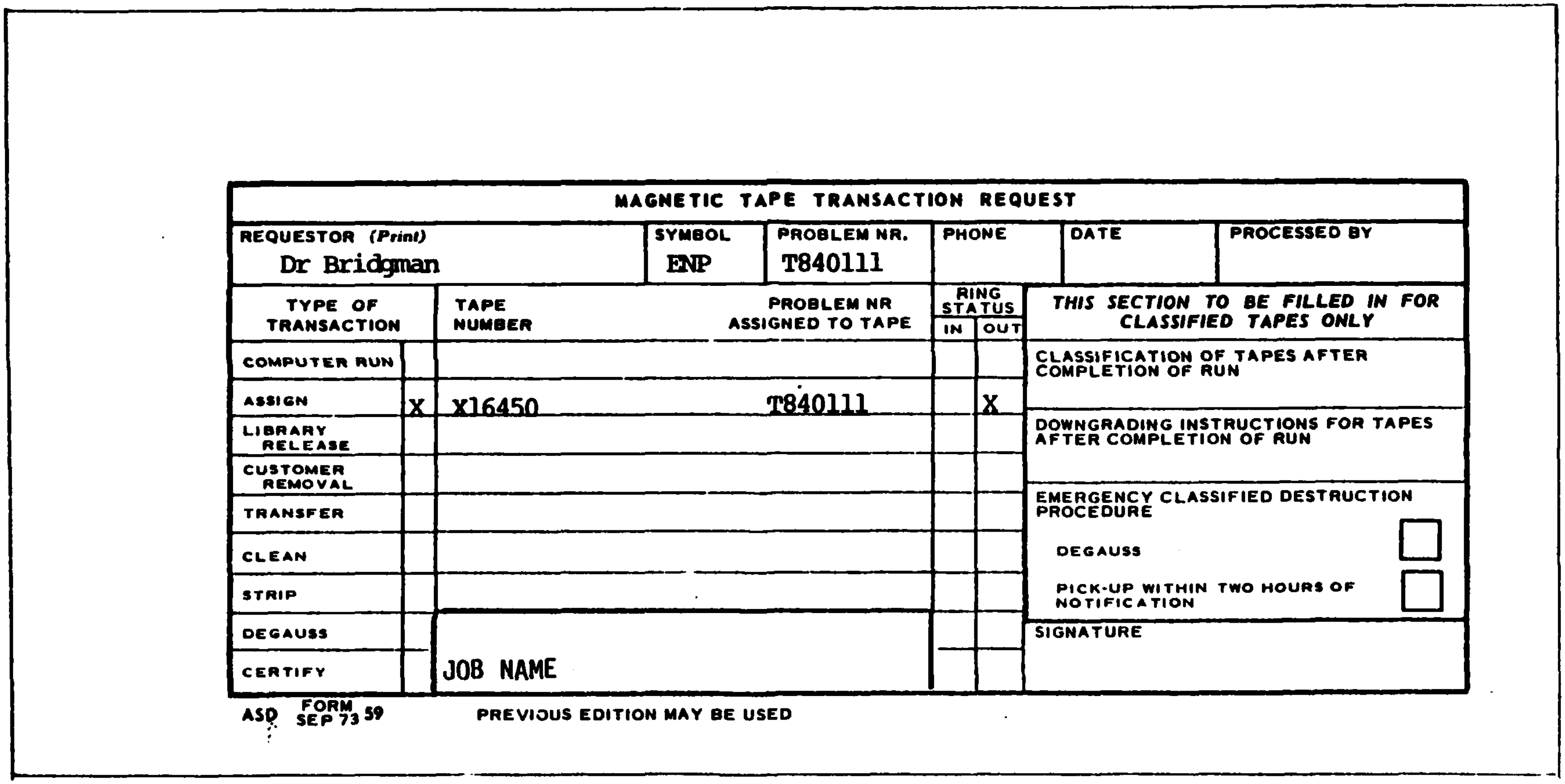
(1)The two data base magnetic tapes must be individually identified before turning them over to the tape library for processing. Each tape can be identified from a tag attached by AFMPC before shipment to AFIT. For each tape, record the reel number (a six digit integer) and the file ID ("AUTHAFIT" for the Requirements tape and "ASGDAFIT" for the Inventory tape.

Hand carry the two tapes to the control desk at the Aeronautical Systems Division (ASD) Computer Center, Bldg 676. Inform the technician that you need "X" numbers (visual serial numbers) assigned to each tape. The technician will ask you to fill out several forms with your problem number, office symbol, and phone. Then you will attach a white label you filled out to each tape. Be sure to record the "X" number for each tape in some type of a log, as these numbers are needed to run the jobs for building the new data bases. Make sure the correct "X" number is associated with each tape, as the tapes are processed by separate jobs.

(2)The data bases can be built now. The magnetic tape jobs used to build the data bases can be submitted as a card deck using a Magnetic Tape Transaction Request card (ASD Form 59). This card is available at the AFIT School of Engineering Computer Lab, room 133. The form should be filled out as shown in figure 13.

Tape Test

A test run should first be made on each magnetic tape. The card deck contents are shown in figure 14, and actual card decks are contained in the ADRIS operations folder maintained at the AFIT School of Engineering by the ADRIS systems monitor. All cards are punched starting in column 1. To test the SPLY build



MAGNETIC TAPE TRANSACTION REQUEST

REQUESTOR *(Print)*	SYMBOL	PROBLEM NR.	PHONE	DATE	PROCESSED BY
Dr Bridgman	ENP	T840111			

TYPE OF TRANSACTION		TAPE NUMBER	PROBLEM NR ASSIGNED TO TAPE	RING STATUS IN	RING STATUS OUT
COMPUTER RUN					
ASSIGN	X	X16450	T840111		X
LIBRARY RELEASE					
CUSTOMER REMOVAL					
TRANSFER					
CLEAN					
STRIP					
DEGAUSS		JOB NAME			
CERTIFY					

THIS SECTION TO BE FILLED IN FOR CLASSIFIED TAPES ONLY

CLASSIFICATION OF TAPES AFTER COMPLETION OF RUN

DOWNGRADING INSTRUCTIONS FOR TAPES AFTER COMPLETION OF RUN

EMERGENCY CLASSIFIED DESTRUCTION PROCEDURE

DEGAUSS □

PICK-UP WITHIN TWO HOURS OF NOTIFICATION □

SIGNATURE

ASD FORM SEP 73 59 PREVIOUS EDITION MAY BE USED

Figure 13, Magnetic Tape Transaction Request program with the inventory tape, enter the job as described in figure 14. The card deck contents for testing the requirements tape are also shown in figure 14, and this job can be run at the same time as the inventory tape test, if desired. There are four things to look for in the printouts from each of these jobs to see if the tape tests are satisfactory (figure 14 and all subsequent figures that depict card decks and dayfile output from computer runs are examples related to NOS/BE. These jobs and output are expected to change slightly under NOS, which will be fully operational in 1984).

(1)Check the number of records for the test (500 is normally an adequate number). The records should be printed out and there should also be a record storage directory for each job (one for each tape). Check the records to make sure the information printed looks correct. An example of the printed output and storage directory format is contained in figure 15. Note the records are printed in ascending order of the ASCs (all "0" ASCs first, followed by all "1" ASCs, etc). The starting number for each new ASC group should correspond to the equivalent storage directory file value. To help analyze the records, every tenth one is numbered.

(2)Check the SPLY program listing to make sure the tape creation date that was entered in the SPLY run deck was printed.

(3)If a record is found with a bad ASC (non-digit first

INVENTORY TEST

(Submit card deck as follows with ASD Form 59)

```
$JOB H80 SYST BCDDMP PRI=15
*RJE 100 H80 *
BRG1.
USER,T840111,RANALLO.
CHARGE,*.
BEGIN,TSPLY,BUILD,(AFMPC No),(X No).
/*EOR
(Date)
2,500
/*EOR
6&9 Multipunch
```

Notes

1. BRG1 is job identification banner. It appears at the top of the computer printout from the job.

2. T840111 is the account number the job runs under.

3. MPC No is the number assigned to the tape from AFMPC.

4. X No is the number assigned at the ASD computer center.

5. Date means to enter the as of date of the file.

6. 2,500 indicates test run, checking the first 500 records.

REQUIREMENTS TEST

(Submit JOB, RJE cards with ASD form, plus the following)

```
BRG2.
USER,T840111,RANALLO.
CHARGE,*.
BEGIN,TDMND,BUILD,(AFMPC No),(X No).
/*EOR
2,500
/*EOR
6&9 Multipunch
```

Figure 14, Card Deck Jobs for AFMPC Tape Tests

```
EL  ASC   PRE   SUFF  AFSC  CBPO  MAJCOM  GR  COUNT


MS RECORD  1  FOLLOWS:

P   0YKY  G           6611  OD    OS        3
N   0CYY        B     5135  LU    3V        3
Q   0YLC  L           8076  HH    0T        5
     .
     .
     .
                                               10
O   0YKY        E     1525  RX    0M        4
     .
     .
     .



     *** RECORD STORAGE DIRECTORY ***


PS( 1)=        1     INTER-AREA
PS( 2)=       26     ADMIN, MAN MIL SCI
PS( 3)=       74     ARTS, HUMAN, EDUC
PS( 4)=      159     BIOLOG & AGRICUL SCI
PS( 5)=      167     ENGINEERING
PS( 6)=      303     CIVIL LAW
PS( 7)=      303     MATH
PS( 8)=      327     PHYS SCI
PS( 9)=      373     SOC SCI
PS(10)=      501     YYYY ASCS
PS(11)=      501     LAST REC + 1
PS(12)=      501     SHOULD BE = PS(11)
```

Figure 15, Sample Output for Tape Test Run character), this is noted in the output listing and the record is also printed. More than a few errors of this type would indicate the tape format has been changed or there are many errors on the AFMPC file used to produce the tape (this check also applies to the actual data base build runs, too).

(4)If a record has alphabetic characters where numeric characters are expected, the printout will indicate an "illegal data in field" error for each occurrence (up to 50) and point out the offending character. An example would be a nonnumeric character in an AFSC. There should not be more than a few (if any) of these errors for the whole data base. A record with such an error (either in grade or AFSC) will be accepted into the data base (this check also applies to the actual data base build runs, too).

If there is any doubt about the validity of test runs, the ADRIS maintainer should get assistance from AFIT/ADO.

Data Base Creation

The Inventory and Requirements data bases must be created separately. First, submit a card deck as shown in figure 16 to create the Inventory file. The results of this run can be verified only by checking the job dayfile, a summary report of the job, found at the end of the output listing. The dayfile

INVENTORY BUILD

```
(Submit ASD Form 59 followed by)
$JOB H80 SYST BCDDMP PRI=15 OUT=0
*RJE 100 H80 *
BRG3.
USER,T840111,RANALLO.
CHARGE,*.
BEGIN,SPLYGO,BUILD,(AFMPC No),(X No).
/*EOR
(Date)
1,100000
/*EOR
6&9 Multipunch
```

Notes

The 1,100000 indicates a full data base run, not a test run.

CATALOG EXAMPLE FOR INVENTORY BUILD

```
INITIAL CATALOG
CT ID=  T840111 PFN=ADRISINV
CT CY= 001 SN=AFIT    0000112128 WORDS.
CATALOG,TAPE40,ADRISPOINTER,CY=1,RP=999.
INITIAL CATALOG
CT ID=  T840111 PFN=ADRISPOINTER
CT CY= 001 SN=AFIT    0000000128 WORDS.
```

REQUIREMENTS BUILD

```
(JOB,REJ and ASD Form 59 followed by)
BRG4.
USER,T840111,RANALLO.
CHARGE,*.
BEGIN,DMNDGO,BUILD,(AFMPC No),(X No).
/*EOR
1,100000
/*EOR
6&9 Multipunch
```

CATALOG EXAMPLE FOR REQUIREMENTS BUILD

```
INITIAL CATALOG
CT ID= T840111 PFN=ADRISREQ
CT CY=001 SN=AFIT    0000019264 WORDS.
```

Figure 16, Data Base Creation Examples

should contain two successful "initial catalogs", similar to the example shown in figure 16. The size of the pointer file should always be 128 words; however, the ADRISINV file can be expected to change slightly from quarter to quarter.

The printout of the record storage directory above the dayfile should also be checked. The entries can be compared with the previous data base values to determine changes in the size of each ASC group.

The SPLY program will also print out the size of the inventory key index. If the SIZE times 100 (- 500) is less than the last entry in the pointer file, PS(12), notify the ADRIS monitor as this indicates the file size is nearing the limit established for ADRIS and it will need to be changed.

Once the Inventory data base has been successfully created, the card deck for the DMND program can be run (figure 16 also contains the card deck format for this run). The same basic checks performed for the Inventory run can also be done for the Requirements run. An example of the dayfile output for the Requirements run is in figure 16. As with the SPLY program, the pointer file should remain constant at 128 words while the ADRISREQ file should change slightly from quarter to quarter. If all the output seems in order, ADRIS is ready to operate using the newly created Inventory and Requirements files. As a final check, two queries can be run using the select (*) option. One would include the select (*) for ASC and the other would be "N" for ASC. The results of these queries should approximate the following totals from the September 1983 data bases:

Master's Inventory	-	34731
Master's Requirements	-	8643
PHD Inventory	-	978
PHD Requirements	-	857
Bachelor's Inventory	-	19930

The remainder of this Maintenance Guide is for the use of the AFIT/ADO ADRIS monitor. It contains information concerning the programs and data files in ADRIS.

C.2 File Format and Structure

The files associated with ADRIS are two magnetic tapes (received quarterly from AFMPC) containing source information and three files constructed by programs that execute by using these tapes. In addition, there are six data files, separately prepared and maintained, that may require updating as existing data in the files becomes obsolete or new data is added.

Magnetic Tapes

One magnetic tape contains information for the Inventory data base while the other tape contains information for the

Requirements data base. The two tapes are nine-track, 1600 BPI, coded in EBCIDIC, labeled, with 25 records to each block. Record structure and read formatting are shown in table X, and only needed fields are read by the build programs.

Constructed Data Files

The SPLY program builds the Inventory data base while the DMND program builds the Requirements data base. Both data bases are built using the FORTRAN WRITMS statement to create a random file structure which is stored on permanent disc space for interactive program use. Each random record consists of 100 of the tape records. The education level, ASC, AFSC, grade, CBPO and major command of each legal tape record is packed into two words; therefore each CYBER random record is 200 words long.

SPLY and DMND each write a random record pointer index of 12 words to the pointer file. The two data base files are organized by grouping records according to the ASC general area of study (all "0s" together, all "1s" together, etc). Therefore, the pointer index contains the beginning record number of each ASC group.

A third pointer file record is used to store the magnetic tape creation date. This date is printed during use of the interactive ADRIS program.

Table X, Tape and Read Formats

Data Element	Position	Format
INVENTORY: 96 characters per block		
ASC	1-4	4A1
Education Level	5	A1
Duty AFSC Prefix	6	A1
Duty AFSC and Suffix	7-11	I4,A1
Current Grade	12-13	I2
Assignment Availability Date	14-17	I4
PAS CBPO Code	18-19	A2
PAS MAJCOM - ID	20-21	A2
PAS Number	22-25	
Method to Achieve Educational Level	26	
PAS Organization Number	27-30	
PAS Organization Kind	31-33	
PAS Organization Type	34-35	
PAS Installation Name	36-52	
PAS Country or State Name (Abbrev)	53-57	
Functional Account	58-63	
Organizational Structure ID	64-68	
Program Element	69-74	
Restricted Field (Not Used)	75-80	
Blank Fill	81-96	
REQUIREMENTS: 102 characters per block		
ASC	1-4	4A1
Education Level	5	A1
Authorized AFSC Prefix	6	A1
Authorized AFSC and Suffix	7-11	I4,A1
Authorized Grade	12-13	I2
Authorized Manpower Level(15th of month)	14	I1
PAS CBPO Code	15-16	A2
PAS MAJCOM Code	17-18	A2
PAS Number	19-22	
Authorized Functional Acct Descriptor	23-40	
Authorized PAS Organization Number	41-44	
Authorized PAS Organization Kind	45-47	
Authorized PAS Organization Type	48-49	
PAS Installation Name	50-66	
PAS Country or State Name (Abbrev)	67-71	
Avthorized Functional Account	72-77	
Authorized Program Element	78-83	
Authorized Organization Structure ID	84-88	
Blank Fill	89-102	

Auxiliary Files

All six auxiliary data files are sequentially structured. CONVRT is used by programs SPLY and DMND to convert obsolete ASCs to their replacement values. The file contains 30 obsolete ASCs and their replacements. The ASCs are organized in 72 column card-image records. The 30 obsolete ASCs occupy the first 120 character positions with their respective replacement ASCs occupying the corresponding character positiions from 121 to 240. The 30 obsolete ASCs are ordered to reflect a minimum search binary tree structure as explained in chapter 4. If new ASCs are added to the file they should be inserted to maintain the minimum search structure. The build programs expect the first four characters on the file to be the "root" element of the tree.

A new program, TREE, is in the ADRIS program library and it can be used to read data file CONVRT to show the basic tree structure. By using output from this program to manually construct a tree, the ADRIS monitor can determine if a new structure is optimal when new codes are added to the CONVRT file. If the tree is not optimal, the sequence of obsolete codes and their respective replacement codes should be changed. If the number of codes in CONVRT is changed, dimensions in programs SPLY, DMND and TREE must be changed to correspond to the new number of obsolete codes in the file.

GENERAL is used by the DMND program to determine which ASCs must have their third or fourth characters generalized (converted to "Y"). The GENERAL file is used to construct a hash table as explained in chapter 3. File format is 70 column card-image records with seven ASCs and their associated codes. Figure 17 contains an example of this format.

```
::::::4KAC::::B4KXY:::::4LFA.........
```

Figure 17, GENERAL Data File Structure

The file is read with a (5X,R5) format so ":" translates to internal integer 0 and "B" into 2. Code 0 is a cue that the ASC is to be left unchanged while code 2 is a cue to convert the last two characters to "YY". There is no ordering to the file so any additions may be made to the end of the file. Additions should not be made without confirming that no more than two ASCs hash to any particular table position. This can be checked using the HASHTST program stored in the ADRIS program library. HASHTST reads file GENERAL and the output from the program indicates whether or not more than two of the ASCs hash to the same number. If the number of codes in CONVRT changes, statements in programs DMND and HASHTST must be changed to correspond to the new number of codes in the file. Statements in program ADRIS where GENERAL is used for on-line documentation must also be changed.

AREA data file is used by the interactive program to convert AFSC area descriptors (table VII of the User's Guide contains these codes). The file is composed of records as shown by the two examples in figure 18. Characters 1-4 are the area descriptor (four characters must be used)--LOGI would mean logistics and INTE would mean intelligence. Characters 5 and 6 contain the number of constituent AFSCs. Character 7 contains a dash ("-") if the first two AFSCs in the list are to be considered inclusive (i.e., 63xx-66xx), otherwise character 7 is a blank. The remaining characters are the constituent ASCs. When additional area codes are added to the file, statements in program ADRIS (overlay BASIC, in subroutines GTPARAM and GETAFSC) must be changed to reflect the new record size of file AREA.

```
LOGI 9-63XX66XX004X009X31XX40XX46XX60XX0005
INTE 3-80XX57XX0910
```

Figure 18, AREA Data File Structure

AGGREG is used by the interactive program to convert ASC aggregate codes to constituent ASCs. The file is composed of records as shown in figure 19. Characters 1-4 are the aggregate code. Character 5 and 6 represent the number of constituent ASCs, right justified. Positions 7 thru 11 are the ASC indexes (1 for "0" ASCs, 2 for "1" ASCs, etc) for each different group of constituent ASCs. The maximum is five, and the positions are zero filled when there are less than five.

```
AADY06570002010699999905069999994BCY34EDE44IAY3.....
```

Figure 19, AGGREG Data File Structure

Character 12 is the number of different ASC indexes. Positions 13 thru 22 are used to show the number of different ASC indexes (first character) that are associated with each code. For example, if a specific aggregate code had five ASCs beginning with "4" and "4" was the smallest index for this specific code, position 13-14 would be "01" since this is the starting point. Since there are five ASCs for this index, the ending position would be "05" and this would be in positions 23-24, since 23 thru 32 are used to mark the ending count for each index. If the next index was "6" and there were two of these, positions 15-16 would be "06" and 25-26 would be "07". From position 33 on, the constituent ASCs are listed. Each ASC is followed by a digit that indicates the number of specific (non "Y") characters in the ASC. If the number of codes is increased or decreased, statements in program ADRIS (overlay BASIC, subroutine GTPARAM and GTFCT) must be changed to reflect the new number of records in AGGREG.

CBPCODE is used by the interactive program to translate CBPO codes for output reports, and it is also used as documentation that can be printed for the user. Figure 20 contains sample records from this file. The first entry for each record is the translated CBPO name and the second entry in the file is the CBPO code that is used in the ADRIS data bases. If codes are added or deleted to this file, statements in program ADRIS (overlay BASIC, subroutine GTPARAM, and overlay SUMRY) must be changed to reflect the new number of codes in CBPCODE.

```
AFIT      WY
ALCONBURY AH
ALTUS     AM
```

Figure 20, CBPCODE Data File Structure

MAJCODE is used by the interactive program to translate major command codes for output reports, and it is also used as documentation that can be printed for the user. Figure 21 contains sample records from this file. The first entry for each record is the translated major command name, the second entry is the abbreviated command name used in the reports, and the third is the command code abbreviation used in the ADRIS data bases. If codes are added or deleted, statements in program ADRIS (overlay BASIC, subroutine GTPARAM, and overlay SUMRY) must be changed to represent the new number of codes in CBPCODE.

```
AEROSPACE DEFENSE CENTER     ADZ      0Z
AF ACCT/FINANCE CENTER       AFAFC    0E
STRATEGIC AIR COMMAND        SAC      0S
```

Figure 21, MAJCODE Data File Structure

C.3 File Directory

The ADRIS build and interactive programs are stored on permanent file disk space in object form. All ADRIS data files are stored as permanent files and source programs are also stored as permanent files in the same library. Table XI contains a description of all files catalogued in the ADRIS library.

All permanent files are stored under the problem number T840111, which is an account number established under the Network Operating System (NOS). This account number is protected from file expiration and all files are permanently catalogued under the account number.

C.4 Miscellaneous Notes

The random file key indexes (array KEYS in SPLY and ADRIS, array KEYD in DMND and ADRIS) must be kept at least one larger than the number of random records. If the dimensions need to be changed due to increases in the number of records on file, they must be changed in these programs. The COMMON and OPENMS statements in program ADRIS are affected.

Table XI, ADRIS Program Library

FILE NAME	DESCRIPTION
ADRIS	Source for user queries
INVTORY	Inventory data base
ADRSOBJ	Object of interactive ADRIS
POINTER	Pointer file for Inventory and Requirements Data Bases
REQMENT	Requirements data base
AGGREG	Aggregate ASC data file
AREA	Career Area AFSC data file
BUILD	Contains procedures TSPLY, TDMND, SPLYGO, DMNDGO used to test tapes and build new data bases
CBPCODE	CBPO data file
CONVRT	Obsolete/replacement ASC data file
DMNDOBJ	Object for building Requirements file
DMND	Source for Requirements build
GENERAL	ASC generalization data file
GRAN	Procedure to attach files and execute ADRSOBJ. Procedure name is AFIT
HASHTST	Source for hash algorithm testing
MAJCODE	Major command data file
SPLYOBJ	Object for building Inventory file
SPLY	Source for Inventory build
TREE	Source for binary tree testing

A more comprehensive check of the build programs can be accomplished by separately reading the records from the magnetic tape and comparing these records with those printed by the build test programs.

CYBER Record Manager cannot process magnetic tape blocks larger than 5120 characters.

The ADRIS interactive program expects the following data file assignments:

TAPE1 - Inventory data base

TAPE2 - Requirements data base

TAPE4 - Pointer file

TAPE6 - Aggregate data file

TAPE7 - CBPO code data file

TAPE8 - Major command data file

TAPE9 - Area data file

TAPE12 - General data file

The object code overlay of the ADRIS interactive program expects the overlays to be stored on file AADMS.

VITA

Gene Patrick Ranallo was born in Akron, Ohio on 23 August 1946. He entered the Air Force in September 1966 and later attended Oklahoma State University in January 1969. He graduated in January 1971 with a Bachelor of Science Degree in Business, majoring in Electronic Data Processing, and was commissioned at the Officer Training School at Lackland AFB in April 1971. He served as a personnel officer at Charleston AFS, Maine and at England AFB, Louisiana prior to being transferred to the Air Force Manpower and Personnel Center (AFMPC), Randolph AFB, Texas in March 1976. After serving over four years as a personnel data systems analyst, he was assigned in June 1980 as Chief, Consolidated Base Personnel Office at Lindsey AS, Germany. In June 1982 he began his studies towards a Master's Degree in Information Systems at the Air Force Institute of Technology (AFIT) at Wright-Patterson AFB, Ohio.

Permanent Address: 7418 Hidden Oak
San Antonio, Texas 78233

8a. NAME OF FUNDING/SPONSORING ORGANIZATION	8b. OFFICE SYMBOL *(If applicable)*	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c. ADDRESS *(City, State and ZIP Code)*

10. SOURCE OF FUNDING NOS.

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.

11. TITLE *(Include Security Classification)*
See Box 19

12. PERSONAL AUTHOR(S)
Gene P. Ranallo, B.S., Major, USAF

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT *(Yr., Mo., Day)*	15. PAGE COUNT
MS Thesis	FROM ________ TO ________	1983 December	126

16. SUPPLEMENTARY NOTATION

Approved for public release: IAW AFR 190-17.
LYNN E. WOLAVER 7 Feb 84
Dean for Research and Professional Development
Air Force Institute of Technology (ATC)
Wright-Patterson AFB OH 45433

17. COSATI CODES		
FIELD	GROUP	SUB. GR.
5	1	
9	2	

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*

Advanced Academic Degree(AAD), CDC CYBER, Network Operating System(NOS)

The Advanced Degree Requirements Information System (ADRIS), an interactive data retrieval system that resides on the CDC CYBER 74 computer system, was updated and enhanced. The system provides AFIT staff and faculty members information pertaining to Advanced Academic Degree job positions in the Air Force and to officers who possess advanced degrees. Changes were made to ADRIS to make it easier to use, to provide much needed enhancements, and to add Bachelor's Degree information related to Air Force officers.

Extensive testing was performed throughout the systems modification, and two primary users of the system were involved in evaluating the changes. The operational system was used as a basis for comparison tests to insure tally information in reports was accurate.

The program library containing all source code, data files, and procedure files was restructured to make it easier to use during future system enhancements or maintenance. A system user's guide and maintenance guide were revised to reflect all changes made to the system and to provide additional information not previously documented.